IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

JUNIPER NETWORKS, INC., a Delaware)
corporation,)
)
Plaintiff,)
)
vs.) Civil Action No. 11-1258-SLR
)
PALO ALTO NETWORKS, INC., a Delaware)
corporation,) DEMAND FOR JURY TRIAL
)
Defendant.)

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Juniper Networks, Inc. ("Juniper") hereby asserts the following claims for patent infringement against Defendant Palo Alto Networks, Inc. ("PAN"), and alleges as follows:

NATURE OF ACTION

1. Plaintiff Juniper is a leader in computer networking. In particular, Juniper has pioneered innovations in firewall technology, an integral component to safe and secure computer networking. Firewalls are designed to permit or deny network transmissions based upon a set of rules, and are frequently used to protect networks from unauthorized access while permitting legitimate communications to pass. As such, firewalls are critical to running secure networks.

2. Juniper has developed and owns significant intellectual property relating to firewall technology, including the seven patents at issue in this action: U.S. Patent No. 8,077,723; U.S. Patent No. 7,779,459; U.S. Patent No. 7,650,634; U.S. Patent No. 7,302,700; U.S. Patent No. 6,772,347; U.S. Patent No. 7,734,752; and U.S. Patent No. 7,107,612 (the "patents-in-suit"), all of which relate to core aspects of firewall technology. The innovations of these patents are, among other things, important for efficiently protecting computer networks

from dangerous incoming communications and from individuals attempting to gain unauthorized access to the computer networks.

3. This is a civil action for the willful infringement of the patents-in-suit by Defendant PAN, a company founded by several former high-level employees of Juniper to compete against Juniper. As detailed further below, PAN's founders and key executives include Nir Zuk and Yuming Mao. Zuk and Mao are acutely aware of the patents-in-suit and the significance of the patented inventions to firewall technology because Zuk and Mao personally worked on the technology and participated in prosecution of the patents-in-suit on which they are named inventors when they were executives at NetScreen Technologies, Inc. ("NetScreen"), the predecessor to Juniper's current security business unit.

4. In the largest acquisition in its history, Juniper paid approximately $4 billion to acquire NetScreen and its intellectual property, personally enriching Zuk and Mao through their equity interests in NetScreen. Shortly thereafter, Zuk and Mao left to form PAN as a competitor to Juniper. Zuk and Mao then incorporated into PAN's products the very technologies they learned about—and helped to develop and patent—while at NetScreen and Juniper.

5. PAN now has begun to use Juniper's own patented technology to compete against Juniper, and is publicly claiming that it plans to grow at a rapid pace in markets pioneered by Juniper products.

THE PARTIES

6. Plaintiff Juniper is a corporation organized and existing under the laws of the State of Delaware, with customers throughout the United States including many incorporated

in this judicial district, and with its corporate headquarters located at 1194 North Mathilda Avenue, Sunnyvale, California.

7. Defendant PAN is a corporation organized and existing under the laws of the State of Delaware, with customers throughout the United States including many incorporated in this judicial district, and with its corporate headquarters located at 3300 Olcott Street, Santa Clara CA.

JURISDICTION AND VENUE

8. This is an action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 100 *et seq.*, including 35 U.S.C. § 271.

9. This Court has subject matter jurisdiction over this action under 28 U.S.C. § 1331 and § 1338(a).

10. This Court has personal jurisdiction over Defendant PAN because, among other reasons, PAN is incorporated in Delaware, and has availed itself of the benefits and protections of Delaware law.

11. Venue is proper in this judicial district pursuant to 28 U.S.C. § 1391(b)-(c) and 1400(b), because, among other reasons, Defendant PAN is subject to personal jurisdiction in this judicial district and PAN is incorporated in Delaware.

FACTUAL BACKGROUND

12. Juniper was founded in 1996 to design, develop, and sell innovative high-performance network infrastructure products. Juniper offers its customers a broad product portfolio that spans routing, switching, security, application acceleration, and identity policy and control. These products are designed to provide data security, performance, choice, and flexibility while reducing overall total costs of running computer networks. In just 15 years,

Juniper has become a leader in secure and efficient networking. In addition, through strong industry partnerships, Juniper has fostered path-breaking innovation in the field of computer networks.

13. Juniper's growth and success have been driven by innovative development as well as by intelligent acquisitions. The largest acquisition in Juniper's history took place on April 16, 2004, when it acquired NetScreen for approximately $4 billion. In connection with this acquisition, Juniper acquired NetScreen's intellectual property. At the time, NetScreen was regarded as an industry innovator that had successfully developed high-end network security devices for enterprise and mid-sized companies. An important part of the NetScreen intellectual property portfolio that Juniper acquired was the then-pending NetScreen patent applications, including several that matured into the patents-in-suit. Zuk, the former Chief Technology Officer of NetScreen, and Mao, a former engineering architect at NetScreen, helped develop—and, in fact, are named inventors or co-inventors on—the patents-in-suit.

14. NetScreen was acquired by Juniper in 2004. After NetScreen became part of Juniper, Zuk and Mao, respectively, served as Juniper's Chief Security Technologist and Chief Architect of security products. Less than two years after the acquisition, however, they left Juniper to form PAN. Zuk and Mao subsequently recruited other Juniper employees to help them in their efforts to turn PAN into a competitor to Juniper. Recently, PAN has relied increasingly on former Juniper employees to sell its products that are powered by Juniper's intellectual property.

15. At the time that Zuk and Mao decided to build a new company based on Juniper's technology, they were well aware of the patents-in-suit upon which they were named inventors, and which they knew were part of the valuable intellectual property that Juniper

acquired when NetScreen became part of Juniper. Zuk and Mao nonetheless made the decision to develop and market products at PAN that practice the very inventions to which they were exposed, and indeed helped develop, when they worked for NetScreen and Juniper.

FIRST CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 8,077,723

16. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

17. United States Patent No. 8,077,723 (the "'723 patent"), entitled, "Packet Processing in a Multiple Processor System," was duly and legally issued on December 13, 2011. Juniper is the owner by assignment of all rights to, title to, and interest in the '723 patent. A copy of the '723 patent is attached as Exhibit A.

18. PAN has infringed and is currently infringing the '723 patent in violation of 35 U.S.C. § 271, by making, using, selling and/or offering for sale products that infringe the '723 patent, including PAN's PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

19. PAN has also infringed and is infringing the '723 patent by actively inducing infringement of the '723 patent and/or contributorily infringing the '723 patent. PAN has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the '723 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the '723 patent and intended that third parties using those products would infringe the '723 patent. One fact, among others, that evidences PAN's knowledge and

intent is that Nir Zuk and Yuming Mao, co-founders of PAN, are the inventors of the ‘723 patent.

20. PAN’s acts of infringement of the ‘723 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘723 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

21. PAN’s infringement of the ‘723 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

SECOND CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 7,779,459

22. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

23. United States Patent No. 7,779,459 (the “‘459 patent”), entitled, “Method and Apparatus for Implementing a Layer 3/Layer 7 Firewall in an L2 Device,” was duly and legally issued on August 17, 2010. Juniper is the owner by assignment of all rights to, title to, and interest in the ‘459 patent. A copy of the ‘459 patent is attached as Exhibit B.

24. PAN has infringed and is currently infringing the ‘459 patent in violation of 35 U.S.C. § 271, by making, using selling and/or offering for sale products that infringe the ‘459 patent, including PAN’s PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

25. PAN has also infringed and is infringing the ‘459 patent by actively inducing infringement of the ‘459 patent and/or contributorily infringing the ‘459 patent. PAN

has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the ‘459 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the ‘459 patent and intended that third parties using those products would infringe the ‘459 patent. One fact, among others, that evidences PAN’s knowledge and intent is that Yuming Mao, co-founder of PAN, was an inventor of the ‘459 patent.

26. PAN’s acts of infringement of the ‘459 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘459 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

27. PAN’s infringement of the ‘459 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

THIRD CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 7,650,634

28. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

29. United States Patent No. 7,650,634 (the “‘634 patent”), entitled, “Intelligent Integrated Network Security Device,” was duly and legally issued on January 19, 2010. Juniper is the owner by assignment of all rights to, title to, and interest in the ‘634 patent. A copy of the ‘634 patent is attached as Exhibit C.

30. PAN has infringed and is currently infringing the ‘634 patent in violation of 35 U.S.C. § 271, by making, using selling and/or offering for sale products that infringe the

‘634 patent, including PAN’s PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

31. PAN has also infringed and is infringing the ‘634 patent by actively inducing infringement of the ‘634 patent and/or contributorily infringing the ‘634 patent. PAN has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the ‘634 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the ‘634 patent and intended that third parties using those products would infringe the ‘634 patent. One fact, among others, that evidences PAN’s knowledge and intent is that Nir Zuk, co-founder of PAN, was the inventor of the ‘634 patent.

32. PAN’s acts of infringement of the ‘634 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘634 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

33. PAN’s infringement of the ‘634 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

FOURTH CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 7,302,700

34. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

35. United States Patent No. 7,302,700 (the “‘700 patent”), entitled, “Method and Apparatus for Implementing a Layer 3/Layer 7 Firewall in an L2 Device,” was duly and

legally issued on November 27, 2007. Juniper is the owner by assignment of all rights to, title to, and interest in the ‘700 patent. A copy of the ‘700 patent is attached as Exhibit D.

36. PAN has infringed and is currently infringing the ‘700 patent in violation of 35 U.S.C. § 271, by making, using selling and/or offering for sale products that infringe the ‘700 patent, including PAN’s PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

37. PAN has also infringed and is infringing the ‘700 patent by actively inducing infringement of the ‘700 patent and/or contributorily infringing the ‘700 patent. PAN has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the ‘700 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the ‘700 patent and intended that third parties using those products would infringe the ‘700 patent. One fact, among others, that evidences PAN’s knowledge and intent is that Yuming Mao, co-founder of PAN, was an inventor of the ‘700 patent.

38. PAN’s acts of infringement of the ‘700 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘700 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

39. PAN’s infringement of the ‘700 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

FIFTH CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 6,772,347

40. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

41. United States Patent No. 6,772,347 (the "'347 patent"), entitled, "Method, Apparatus and Computer Program Product for a Network Firewall," was duly and legally issued on August 3, 2004. Juniper is the owner by assignment of all rights to, title to, and interest in the '347 patent. A copy of the '347 patent is attached as Exhibit E.

42. PAN has infringed and is currently infringing the '347 patent in violation of 35 U.S.C. § 271, by making, using, selling and/or offering for sale products that infringe the '347 patent, including PAN's PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

43. PAN has also infringed and is infringing the '347 patent by actively inducing infringement of the '347 patent and/or contributorily infringing the '347 patent. PAN has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the '347 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the '347 patent and intended that third parties using those products would infringe the '347 patent. One fact, among others, that evidences PAN's knowledge and intent is that Yuming Mao, co-founder of PAN, was an inventor of the '347 patent.

44. PAN's acts of infringement of the '347 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from

continuing infringement of the ‘347 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

45. PAN’s infringement of the ‘347 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

SIXTH CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 7,734,752

46. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

47. United States Patent No. 7,734,752 (the “‘752 patent”), entitled, “Intelligent Integrated Network Security Device for High-Availability Applications,” was duly and legally issued on January 8, 2010. Juniper is the owner by assignment of all rights to, title to, and interest in the ‘752 patent. A copy of the ‘752 patent is attached as Exhibit F.

48. PAN has infringed and is currently infringing the ‘752 patent in violation of 35 U.S.C. § 271, by making, using, selling and/or offering for sale products that infringe the ‘752 patent, including PAN’s PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, and PA-500 Firewall.

49. PAN has also infringed and is infringing the ‘752 patent by actively inducing infringement of the ‘752 patent and/or contributorily infringing the ‘752 patent. PAN has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the ‘752 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the ‘752 patent and intended that third parties using those products

would infringe the ‘752 patent. One fact, among others, that evidences PAN’s knowledge and intent is that Nir Zuk and Yuming Mao, co-founders of PAN, are inventors of the ‘752 patent.

50. PAN’s acts of infringement of the ‘752 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘752 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

51. PAN’s infringement of the ‘752 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

SEVENTH CLAIM FOR PATENT INFRINGEMENT

Infringement of U.S. Patent No. 7,107,612

52. The allegations in the foregoing paragraphs of this Complaint are incorporated by reference herein as if restated and set forth in full.

53. United States Patent No. 7,107,612 (the “‘612 patent”), entitled, “Method, Apparatus and Computer Program Product for a Network Firewall,” was duly and legally issued on September 12, 2006. Juniper is the owner by assignment of all rights to, title to, and interest in the ‘612 patent. A copy of the ‘612 patent is attached as Exhibit G.

54. PAN has infringed and is currently infringing the ‘612 patent in violation of 35 U.S.C. § 271, by making, using, selling and/or offering for sale products that infringe the ‘612 patent, including PAN’s PA-5000 Series Firewalls, PA-4000 Series Firewalls, PA-2000 Series Firewalls, PA-500 Firewall, and PA-200 Firewall.

55. PAN has also infringed and is infringing the ‘612 patent by actively inducing infringement of the ‘612 patent and/or contributorily infringing the ‘612 patent. PAN

has sold, caused to be sold, or offered to sell, through intermediaries, as an intermediary, or otherwise, the above-referenced products to third parties that then used the products to infringe the ‘612 patent. These products have no substantial non-infringing use. Moreover, PAN knew that these products infringed the ‘612 patent and intended that third parties using those products would infringe the ‘612 patent. One fact, among others, that evidences PAN’s knowledge and intent is that Yuming Mao, co-founder of PAN, is an inventor of the ‘612 patent.

56. PAN’s acts of infringement of the ‘612 patent have caused damage to Juniper, and Juniper is entitled to recover damages from PAN. Moreover, unless enjoined from continuing infringement of the ‘612 patent, PAN will continue to harm Juniper’s interests, causing Juniper irreparable injury as a result of PAN’s conduct.

57. PAN’s infringement of the ‘612 patent has been and continues to be willful and deliberate, entitling Juniper to increased damages under 35 U.S.C. § 284 and reasonable attorneys’ fees under 35 U.S.C. § 285.

PRAYER FOR RELIEF

Wherefore, Plaintiff Juniper prays for judgment as follows:

58. That PAN has infringed, induced infringement of, and/or contributorily infringed one or more of the claims of each of the patents-in-suit;

59. That PAN and its affiliates, subsidiaries, directors, officers, employees, attorneys, agents, and all persons in active concert or participation with any of them be preliminarily and permanently enjoined from further acts of infringement, inducing infringement, and/or contributory infringement of the patents-in-suit;

60. That PAN pay Juniper damages which in no event shall be less than a reasonable royalty, together with interest and costs under 35 U.S.C. § 284;

61. That PAN be ordered to provide an accounting;

62. That PAN's infringement has been willful and that the damages will be increased under 35 U.S.C. § 284 to three times the amount found or measured;

63. That this be adjudged an exceptional case and that Juniper be awarded its reasonable attorneys' fees under 35 U.S.C. § 285;

64. That PAN be required to pay pre- and post-judgment interest on the assessed damages; and

65. That Juniper be awarded any other and further relief as this Court deems just and proper.

DEMAND FOR JURY TRIAL

Juniper hereby demands a trial by jury on all issues so triable.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

/s/ Jennifer Ying

Jack B. Blumenfeld (#1014)
Jennifer Ying (#5550)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19801
(302) 658-9200
jblumenfeld@mnat.com
jying@mnat.com

Attorneys for Plaintiff

OF COUNSEL:

Morgan Chu
Jonathan S. Kagan
Lisa S. Glasser
David C. McPhie
Rebecca Clifford
IRELL & MANELLA LLP
1800 Avenue of the Stars
Suite 900
Los Angeles, CA 90067-4276
(310) 277-1010

September 21, 2012
6466326

EXHIBIT A

US008077723B2

(12) **United States Patent**
Zuk et al.

(10) **Patent No.: US 8,077,723 B2**
(45) **Date of Patent: *Dec. 13, 2011**

(54) **PACKET PROCESSING IN A MULTIPLE PROCESSOR SYSTEM**

(75) Inventors: **Nir Zuk**, Palo Alto, CA (US); **Yu Ming Mao**, Milpitas, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/780,695**

(22) Filed: **May 14, 2010**
(Under 37 CFR 1.47)

(65) **Prior Publication Data**

US 2010/0220727 A1 Sep. 2, 2010

Related U.S. Application Data

(63) Continuation of application No. 11/338,732, filed on Jan. 25, 2006, now Pat. No. 7,746,862.

(60) Provisional application No. 60/704,432, filed on Aug. 2, 2005.

(51) **Int. Cl.**
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **370/392**; 370/401

(58) **Field of Classification Search** 370/389, 370/392, 394, 400, 401, 474
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,598,410	A	1/1997	Stone	
5,606,668	A	2/1997	Shwed	
5,835,726	A	11/1998	Shwed et al.	
6,006,264	A	12/1999	Colby et al.	
6,119,236	A	9/2000	Shipley	
6,253,321	B1	6/2001	Nikander et al.	
6,275,942	B1	8/2001	Bernhard et al.	
6,279,113	B1	8/2001	Vaidya	
6,301,668	B1	10/2001	Gleichauf et al.	
6,304,975	B1	10/2001	Shipley	
6,311,278	B1	10/2001	Raanan et al.	
6,321,338	B1	11/2001	Porras et al.	
6,370,603	B1 *	4/2002	Silverman et al.	710/72
6,421,730	B1 *	7/2002	Narad et al.	709/236
6,449,647	B1	9/2002	Colby et al.	
6,453,345	B2	9/2002	Tracka et al.	
6,466,985	B1	10/2002	Goyal et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

JP	2000-312225	11/2000

(Continued)

OTHER PUBLICATIONS

Co-pending U.S. Appl. No. 11/338,732, filed Jan. 25, 2006 entitled "Packet Processing in a Multiple Processor System" by Nir Zuk et al., 38 pages.

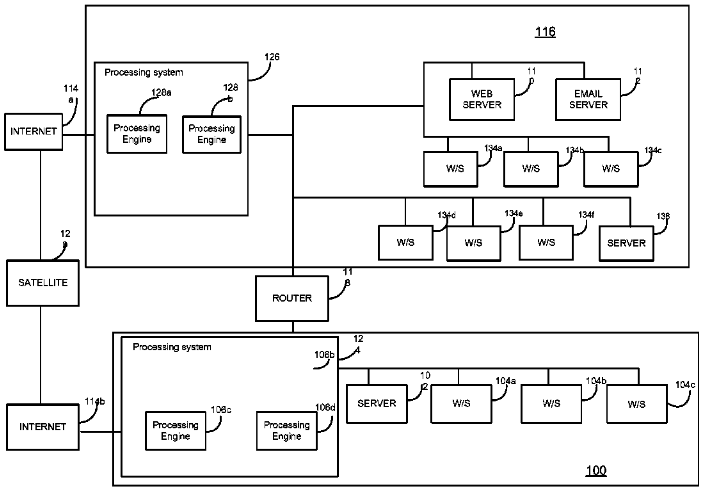
(Continued)

Primary Examiner — Hong Cho
(74) *Attorney, Agent, or Firm* — Harrity & Harrity, LLP

(57) **ABSTRACT**

Packet processing is provided in a multiple processor system including a first processor to processing a packet and to create a tag associated with the packet. The tag includes information about the processing of the packet. A second processor receives the packet subsequent to the first processor and processes the packet using the tag information.

20 Claims, 10 Drawing Sheets



U.S. PATENT DOCUMENTS

6,487,666	B1	11/2002	Shanklin et al.	
6,499,107	B1	12/2002	Gleichauf et al.	
6,600,744	B1 *	7/2003	Carr et al.	370/392
6,768,738	B1	7/2004	Yazaki et al.	
6,781,992	B1 *	8/2004	Rana et al.	370/394
6,788,648	B1	9/2004	Peterson	
6,851,061	B1	2/2005	Holland et al.	
6,856,991	B1	2/2005	Srivastava	
7,006,443	B2	2/2006	Storr	
7,139,679	B1 *	11/2006	McGrew	702/186
7,376,085	B2	5/2008	Yazaki et al.	
7,650,634	B2	1/2010	Zuk	
2001/0028650	A1 *	10/2001	Yoshizawa et al.	370/389
2002/0032797	A1	3/2002	Xu	
2002/0080789	A1 *	6/2002	Henderson et al.	370/392
2002/0124187	A1	9/2002	Lyle et al.	
2002/0126621	A1 *	9/2002	Johnson et al.	370/230
2003/0105976	A1	6/2003	Copeland	
2003/0145225	A1	7/2003	Bruton, III et al.	
2003/0149888	A1	8/2003	Yadav	
2003/0154399	A1	8/2003	Zuk et al.	
2005/0141503	A1 *	6/2005	Welfeld	370/392
2005/0163132	A1 *	7/2005	Mieno et al.	370/395.53
2006/0005231	A1	1/2006	Zuk et al.	

FOREIGN PATENT DOCUMENTS

JP	2003-78549	3/2003
WO	WO 03/025766	3/2003

OTHER PUBLICATIONS

International Search Report for corresponding PCT application, PCT/US2004/009607, dated Oct. 22, 2004, 3 pages.

Stonesoft, "StoneBeat Security Cluster White Paper," Aug. 2000, Finland, pp. 1-9.

Stonesoft, "Secure Highly Available Enterprise—A White Paper," Feb. 2001, Finland, pp. 1-10.

Stonesoft, "StoneGate White Paper," Mar. 2001, Finland, pp. 1-6.

Stonesoft Corp., "StoneGate," product webpage, www.stonesoft.com/document/363.html, Mar. 27, 2001 (print date), pp. 1-2.

Stonesoft Corp., "Next Level of Network Accessibility" webpage, www.stonesoft.com/document/183.html, Mar. 27, 2001 (print date), p. 1.

Stonesoft Corp., "Platforms," webpage, www.stonesoft.com/document/186.html, Mar. 27, 2001 (print date), p. 1.

Nokia, "Technical White Paper: The IP Clustering Power of Nokia VPN—Keeping Customers Connected," Apr. 2001, pp. 1-13.

Nokia, "Nokia VPN Solutions—Nokia VPN CC2500 Gateway," 2001, product information, pp. 1-2.

Nokia, "Nokia VPN Solutions—Nokia VPN CC5200 Gateway," 2001, product information, pp. 1-2.

Nokia, "Nokia VPN Solutions—Nokia VPN CC5205 Gateway," 2001, product information, pp. 1-2.

Axelsson, S., "Intrusion Detection Systems: A Survey and Taxonomy," Dept. of Computer Eng., Chalmers Univ. of Technology, Goteborg, Sweden, Mar. 14, 2000, pp. 1-27.

Avolio, F., "Firewalls and Virtual Private Networks," CSI Firewall Archives, printed Nov. 13, 2001, URL: http://www.spirit.com/CSI/Papers/fw+vpns.html, pp. 1-7.

Bace, R., "An Introduction to Intrusion Detection & Assessment," ICSA Intrusion Detection Systems Consortium White Paper, 1999, URL: http://www.icsalabs.com/html/communities/ids/whitepaper/Intrusion1.pdf, pp. 1-38.

Business Wire, Inc., "NetScreen and OneSecure Unite to Deliver Industry's First Total Managed Security Services Platform," San Jose, CA, Feb. 20, 2001, pp. 1-2.

Business Wire, Inc., "OneSecure Launches the First Co-Managed Security Services Platform," Denver, CO, Jan. 29, 2001, pp. 1-2.

Carr, Jim, "Intrusion Detection Systems: Back to Front?," Network Magazine, Sep. 5, 2001, URL: http://www.networkmagazine.com/article/NMG20010823S0007/2, pp. 1-9.

Check Point Software Technologies Ltd., Firewall-1® Technical Overview P/N 500326, www.checkpoint.com, Oct. 2000, pp. 1-29.

Cisco Systems, "Cisco IOS Firewall Intrusion Detection System," Cisco IOS Release 12.0(5)T, 2001, pp. 1-40.

Cisco Systems, "Cisco IOS Firewall Authentication Proxy," Cisco IOS Release 12.0(5)T, 2001, pp. 1-48.

Clark, D., "RFC815-IP Datagram Reassembly Algorithms," Internet RFC/STD/FYI/BCP Archives, http://www.faqs.org/rfcs/rfc815.html, Jul. 1982, pp. 1-8.

Copeland, Dr. John A., "Observing Network Traffic-Techniques to Sort Out the Good, the Bad, and the Ugly," PowerPoint Slide Presentation presented to ISSA-Atlanta, Jun. 27, 2001, pp. 1-22.

Denning, Dorothy E., "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, No. 2, Feb. 1987, 17 pages.

Farrow, Rik, "An Analysis of Current Firewall Technologies," CSI 1997 Firewalls Matrix, 1998, URL: http://www.spirit.com/CSI/Papers/farrowpa.htm, pp. 1-5.

Firewall Product Comparison Table: VelociRaptor, BorderWare Firewall Server and Firewall-1/VPN-1 Gateway, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: PIX Firewall, CyberGuard Firewall for UnixWare & CyberGuard Firewall for Windows NT, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: CyberGuard Premium Appliance Firewall, InstaGate EX & BizGuardian VPN Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Server Protector 100, GNAT Box Firewall Software & Lucent Managed Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: Internet Security and Acceleration (ISA) Server 2000, NetBSD/i386 Firewall & Guardian Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: NetScreen-10 and NetScreen-100, CyberwallPLUS & BorderManager, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: Gauntlet Firewall, Barricade Classic/XL & Barricade S, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Sidewinder™, SecurePipe Managed Firewall Service & SnapGear, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: SonicWALL PRO, Sunscreen Secure Net & WinRoute Pro 4.1, www.spirit.com printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: WatchGuard Technologies, Inc. LiveSecurity System 4.6, www.spirit.com, printed Nov. 13, 2001, pp. 1-4.

Graham, R., "FAQ: Network Intrusion Detection System," www.robertgraham.com/pubs/network-intrusion-detection.html, Ver. 0.8.3, Mar. 21, 2000, pp. 1-43.

Habra, N. et al., "ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis," Proceedings of the ESORICS '92, European Symposium on Research in Computer Security, Nov. 23-25, 1992, Toulouse, Springer-Verlag, 16 pages.

ICSA Labs, Intrusion Detection System Buyer's Guide, ICSA White Paper, 1999, pp. 1-52.

Jackson, K. et al., "Intrusion Detection System (IDS) Product Survey," Los Alamos National Laboratory, Los Alamos, NM, LA-UR-99-3883 Ver. 2.1, Jun. 25, 1999, pp. 1-103.

Jones, Kyle, "Introduction to Firewalls," IT Audit.org Forum Network Management, vol. 2, May 1, 1999, URL: http://www.itaudit.org/forum/networkmanagement/f209nm.htm, pp. 1-5.

Lancope, "The Security Benefits of a Flow-Based Intrusion Detection System," White Paper, included in IDS filed on Jan. 25, 2006 for U.S. Appl. No. 11/338,732, pp. 1-11.

LapLink, Inc., "Article #178—Introduction to Firewalls," www.laplink.com/support/kb/article.asp?ID=178, Apr. 24, 2001, pp. 1-3.

Mchugh, J. et al., "Defending Yourself: The Role of Intrusion Detection Systems," Software Engineering Institute, IEEE Software Eng., Sep./Oct. 2000, pp. 42-51.

Network ICE Corporation, "Why Firewalls Are Not Enough," at www.networkice.com/products/firewalls.html, 2000, pp. 1-9.

Power, R. et al., "CSI Intrusion Detection System Resource—Five Vendors Answer Some No-Nonsense Questions on IDS," Computer Security Alert #184, Jul. 1998, pp. 1-8.

Power, R., "CSI Roundtable: Experts discuss present and future intrusion detection systems," Computer Security Journal, vol. XIV, #1, URL: http://www.gocsi.com/roundtable.htm, 2001, pp. 1-20.

Sample, Char et al., "Firewall and IDS Shortcomings," SANS Network Security, Monterey, CA, Oct. 2000, pp. 1-13.

Smith, Gary, "A Brief Taxonomy of Firewalls—Great Walls of Fire," SANS Institute's Information Security Reading Room, May 18, 2001, URL: http://www.sans.org/infosecFAQ/firewall/taxonomy.htm, pp. 1-21.

Spitzner, Lance, "How Stateful is Stateful Inspection? Understanding the FW-1 State Table," http://www.enteract.com/~1spitz/fwtable.html, Nov. 29, 2000, pp. 1-8.

Sundaram, A., "An Introduction to Intrusion Detection," www.acm.org/crossroads/xrds2-4/intrus.html, Jan. 23, 2001, pp. 1-12.

Tyson, Jeff, "How Firewalls Work," http://www.howstuffworks.com/firewall.htm/printable, 2001, pp. 1-7.

Xinetica, Ltd., "An Overview of Intrusion Detection Systems," Xinetica White Paper, Nov. 12, 2001 (print date), URL: http://www.xinetica.com/tech_explained/general/ids/wp_ids.html, pp. 1-9.

Zuk, Nir, "Protect Yourself With Firewalls," www.techtv.com, Jul. 12, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3325761,00.html, pp. 1-3.

Zuk, Nir, "How the Code Red Worm Works," www.techtv.com, Sep. 21, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3349133,00.html, pp. 1-2.

Petersen, S. et al., "Web apps pose security threat," ZDNet: Tech Update, Jan. 29, 2001, URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2679177,00.html, pp. 1-3.

Lancope, "StealthWatch Provides Early Detection of the Code Red Worm and its Future Variants," www.stealthwatch.com, included in IDS filed on Jan. 25, 2006 for U.S. Appl. No. 11/338,732, pp. 1-4.

Reavis, J., "Cash and Burn," Jun. 2001, 6 pages.

SOS Corporation, "An Introduction to Firewalls," 1995, URL: http://www.uclan.ac.uk/facs/destech/compute/staff/haroun/FIREWALS.HTM, pp. 1-3.

Morgan, Lisa,"Be Afraid, Be Very Afraid," InternetWeek Intrusion Detection Systems, Jan. 3, 2001, pp. 1-6.

Mullins, Robert, "'Cyber war' raises security concerns," Silicon Valley/San Jose Business Journal, May 11, 2001, pp. 1-4.

James P. Anderson Co., "Computer Security Threat Monitoring and Surveillance," Apr. 15, 1980, 56 pages.

Internet Security Systems, Inc., "REALSECURE™, The RealSecure Advantage," 2001, 2 pages.

Chuvakin, A. et al., "Basic Security Checklist for Home and Office Users," SecurityFocus, Nov. 5, 2001, pp. 1-5.

Network Ice, "SMTP WIZ command," 2001, URL: http://networkice.com/Advice/Intrusions/2001006/default.htm, pp. 1-2.

Bace, R. et al., "NIST Special Publication on Intrusion Detection Systems," National Institute of Standards and Technology Special Publication, 1999, pp. 1-51.

* cited by examiner

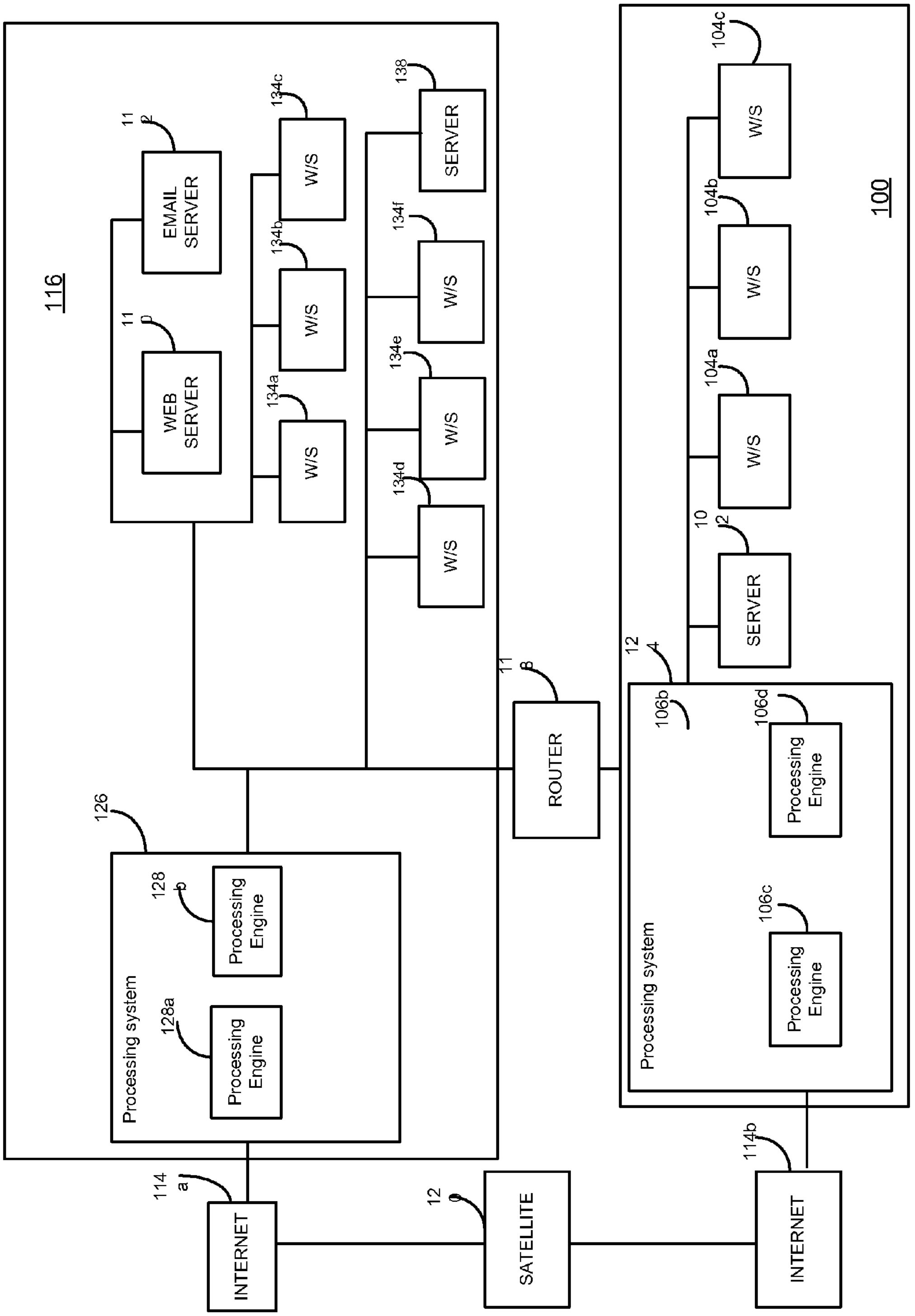

116
WEB SERVER
110
EMAIL SERVER
112
W/S
134a
W/S
134b
W/S
134c
W/S
134d
W/S
134e
W/S
134f
SERVER
138
Processing system
126
Processing Engine
128a
Processing Engine
128b
ROUTER
118
Processing system
124
106b
Processing Engine
106c
Processing Engine
106d
SERVER
102
W/S
104a
W/S
104b
W/S
104c
100
INTERNET
114a
SATELLITE
120
INTERNET
114b

FIG. 1

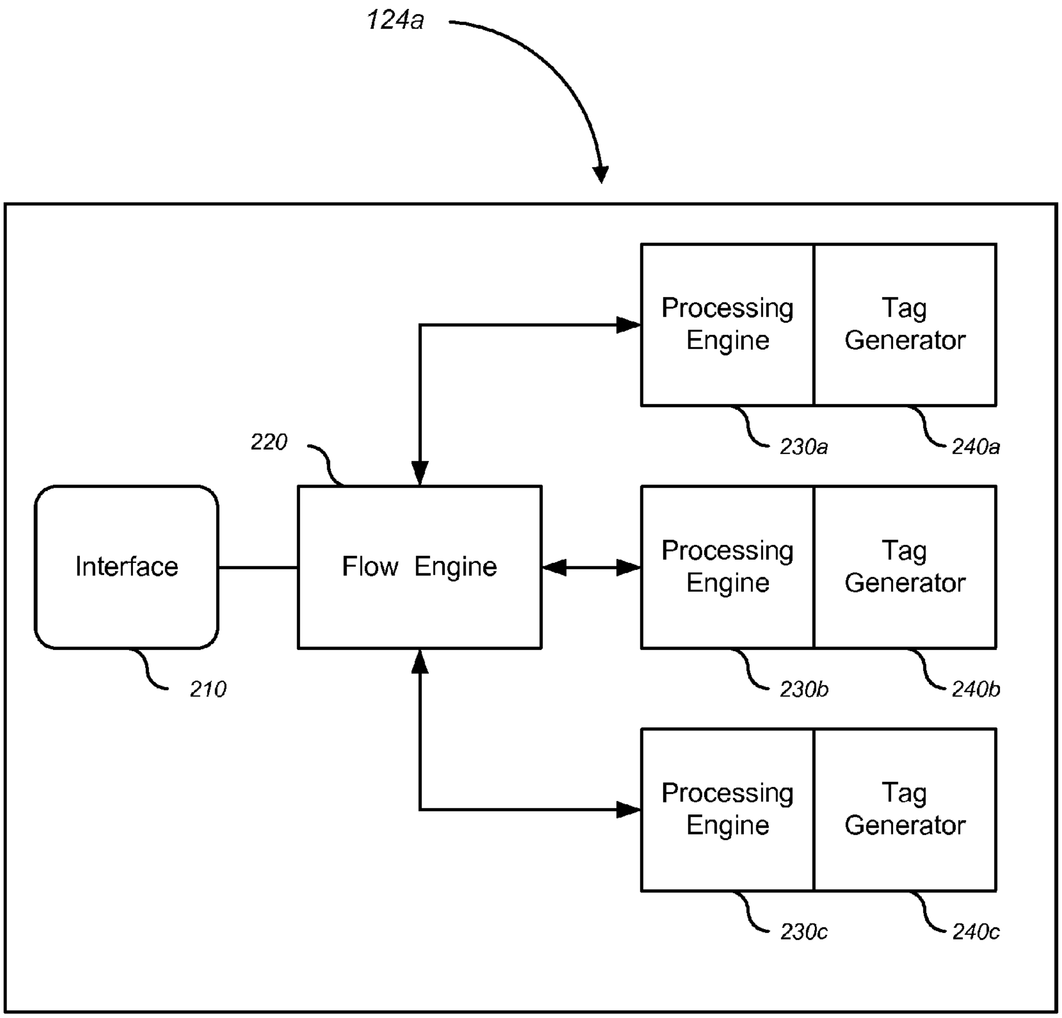

124a
Interface
210
220
Flow Engine
Processing Engine
Tag Generator
230a
240a
Processing Engine
Tag Generator
230b
240b
Processing Engine
Tag Generator
230c
240c

FIG. 2a

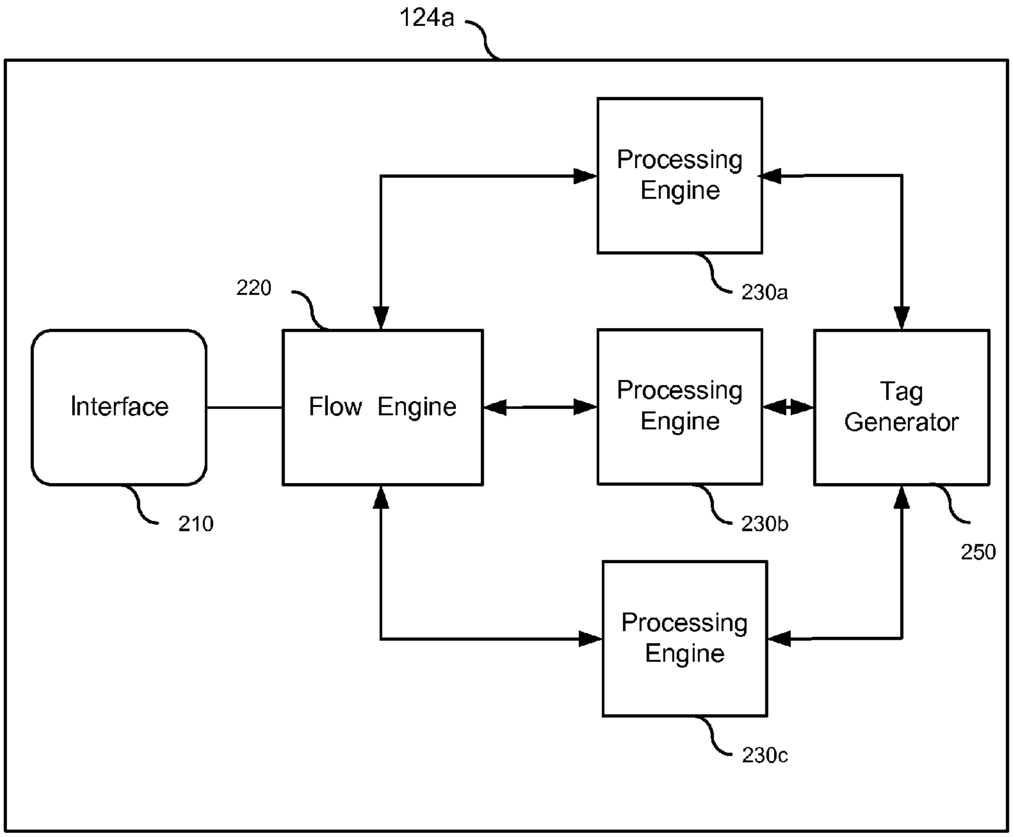

124a
Processing Engine
230a
220
Interface
Flow Engine
Processing Engine
Tag Generator
210
230b
250
Processing Engine
230c

FIG. 2b

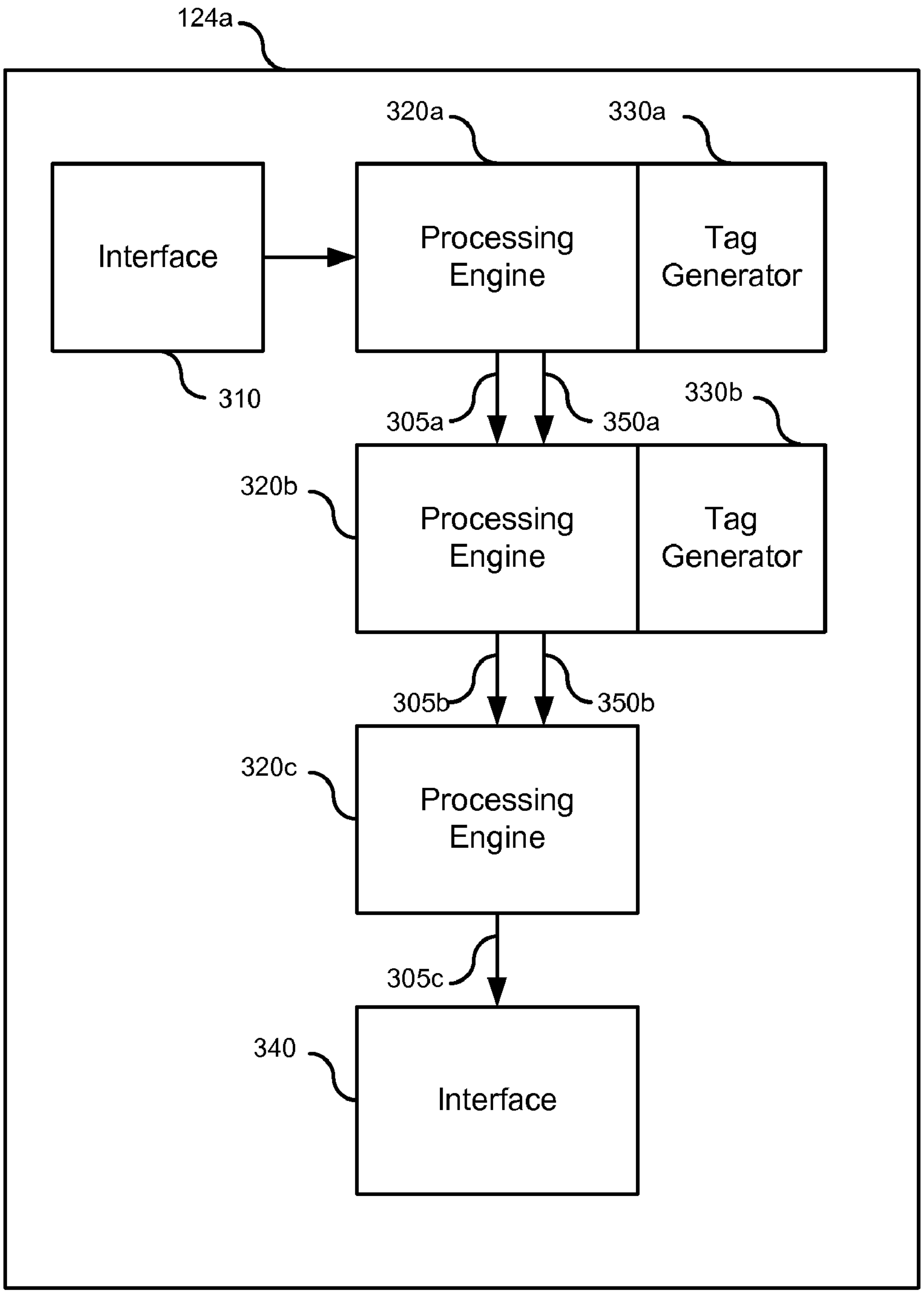

124a
320a
330a
Interface
Processing Engine
Tag Generator
310
305a
350a
330b
320b
Processing Engine
Tag Generator
305b
350b
320c
Processing Engine
305c
340
Interface

FIG. 3a

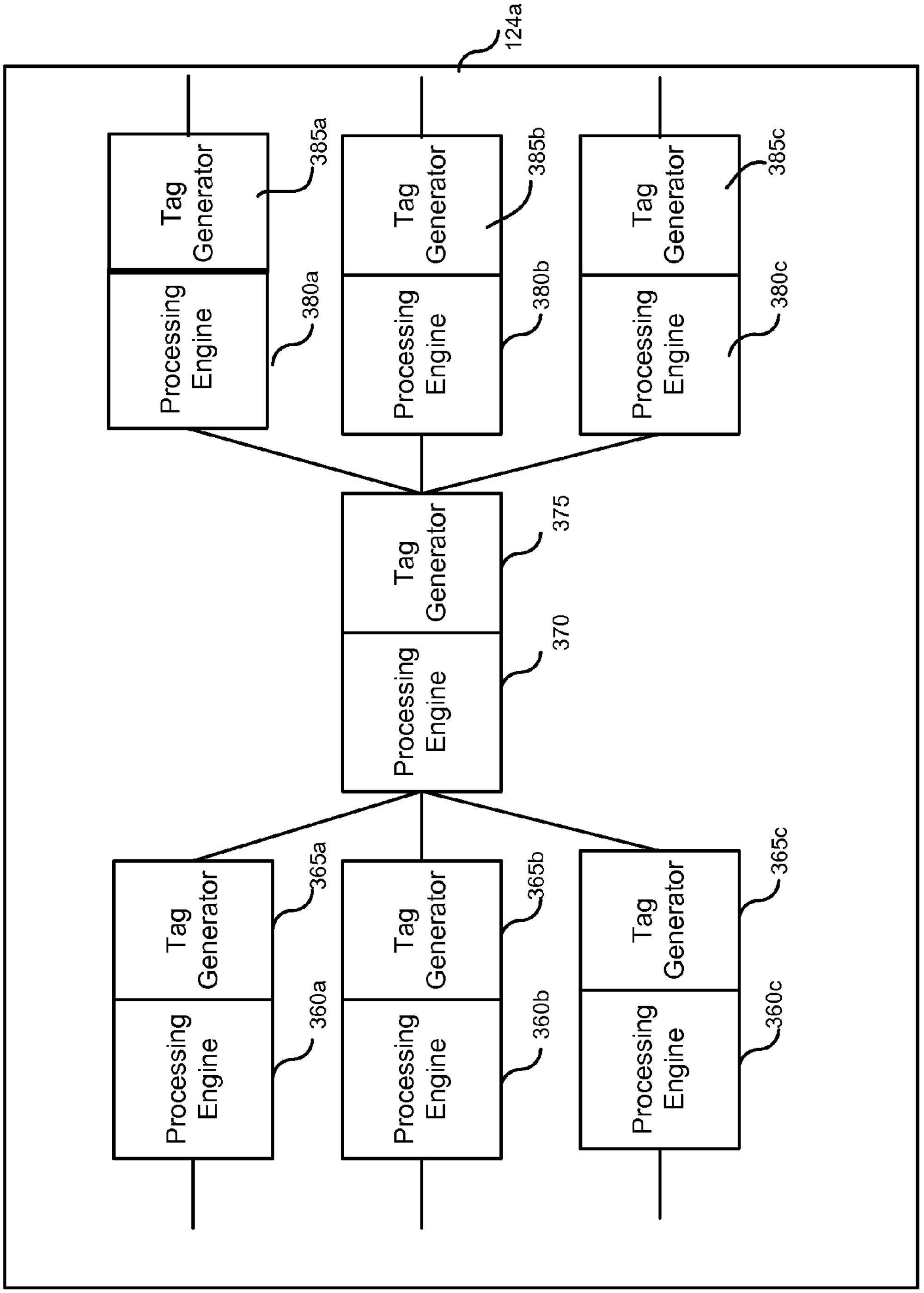

124a
Tag Generator
385a
Processing Engine
380a
Tag Generator
385b
Processing Engine
380b
Tag Generator
385c
Processing Engine
380c
Tag Generator
375
Processing Engine
370
Tag Generator
365a
Processing Engine
360a
Tag Generator
365b
Processing Engine
360b
Tag Generator
365c
Processing Engine
360c

FIG. 3b

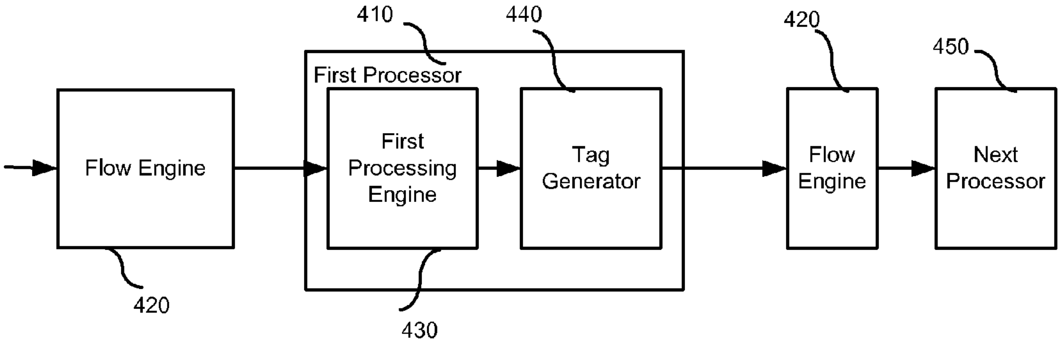

410
440
420
450
First Processor
Flow Engine
First Processing Engine
Tag Generator
Flow Engine
Next Processor
420
430

FIG. 4

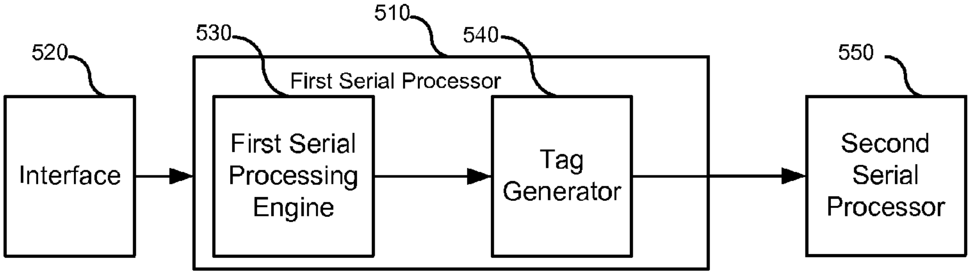

510
530
540
520
550
First Serial Processor
Interface
First Serial Processing Engine
Tag Generator
Second Serial Processor

FIG. 5

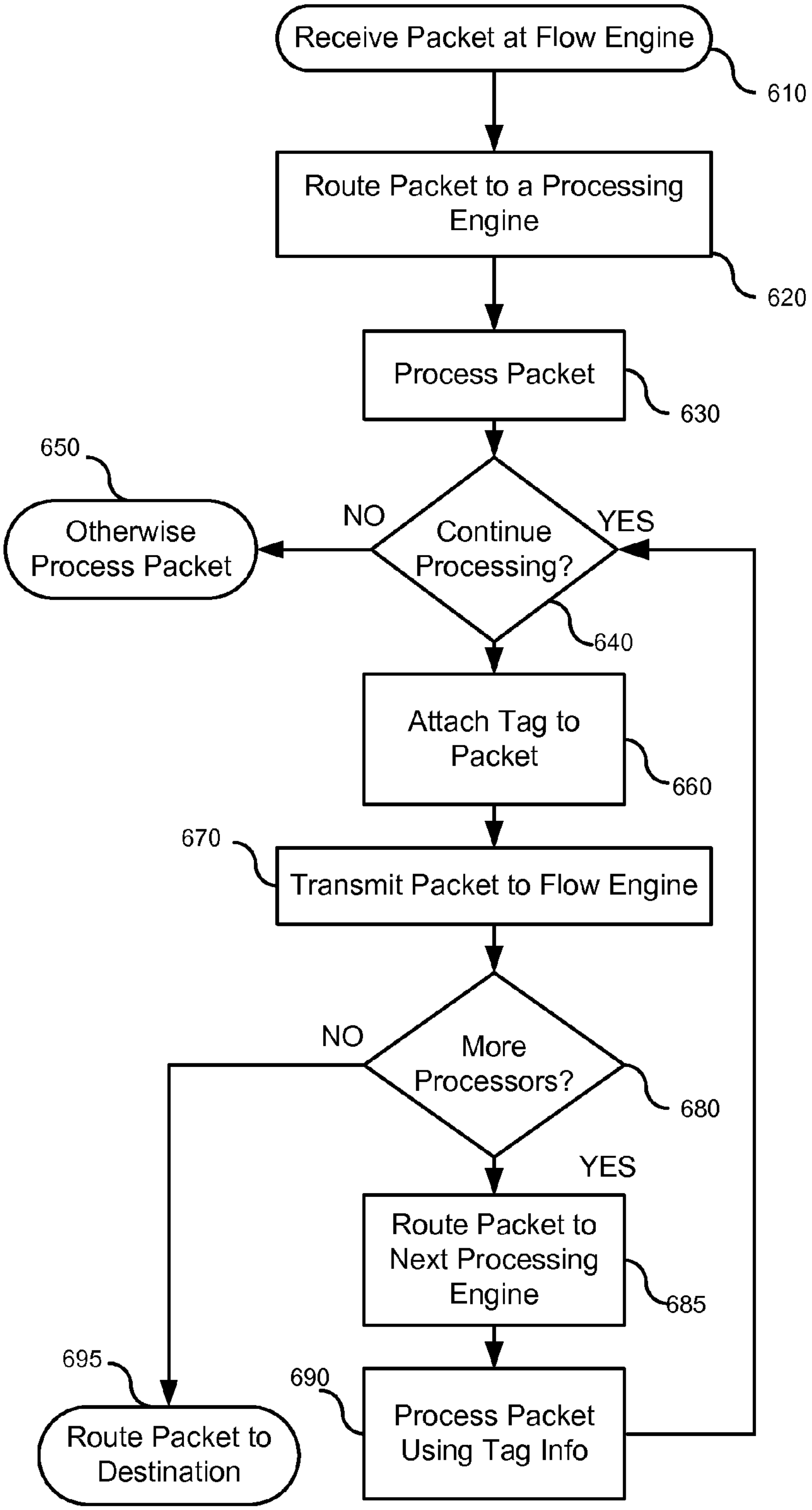

Receive Packet at Flow Engine
610
Route Packet to a Processing Engine
620
Process Packet
630
650
Otherwise Process Packet
NO
Continue Processing?
YES
640
Attach Tag to Packet
660
670
Transmit Packet to Flow Engine
NO
More Processors?
680
YES
Route Packet to Next Processing Engine
685
695
690
Route Packet to Destination
Process Packet Using Tag Info

FIG. 6

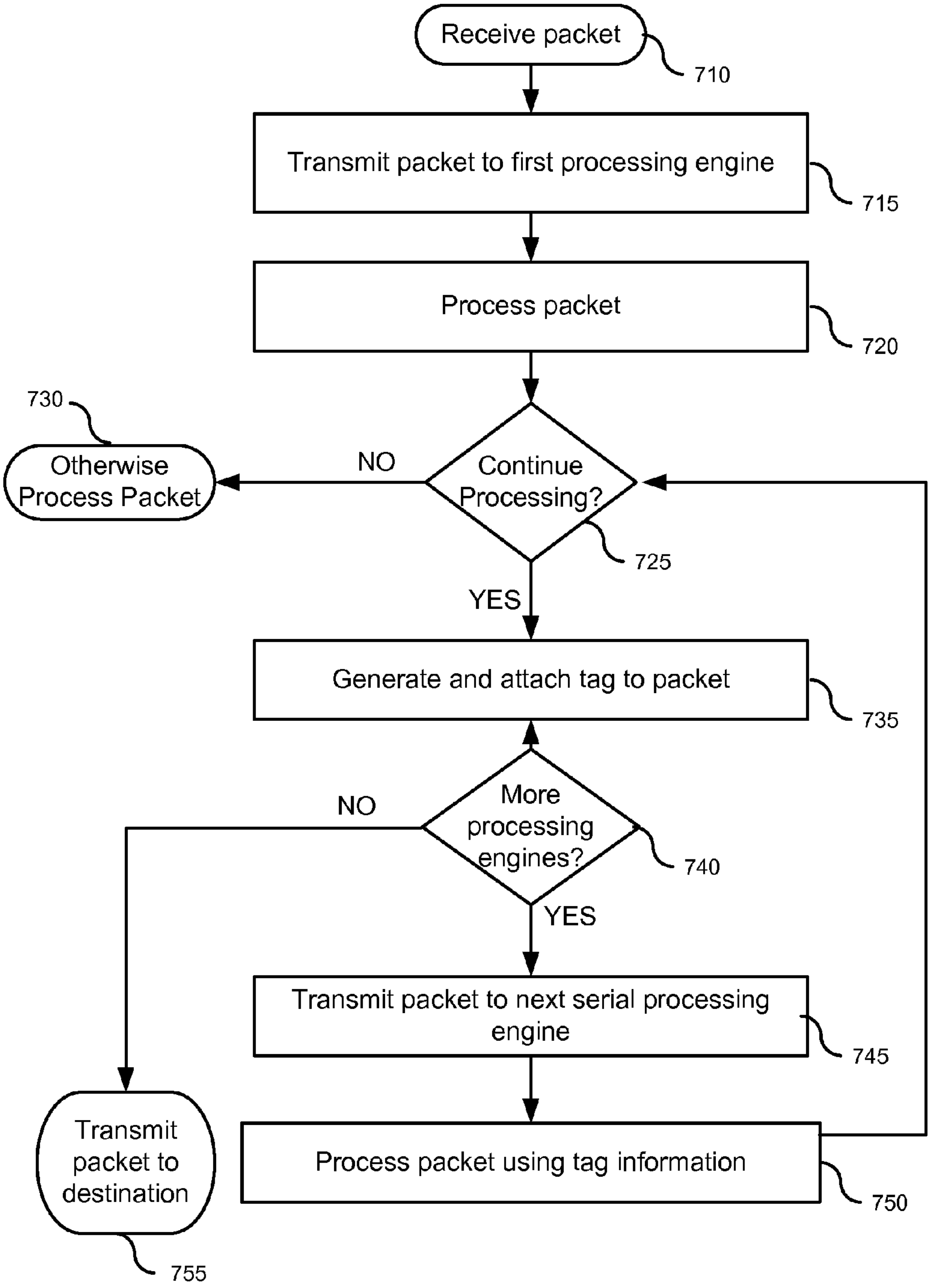

Receive packet
710
Transmit packet to first processing engine
715
Process packet
720
730
Otherwise Process Packet
NO
Continue Processing?
725
YES
Generate and attach tag to packet
735
NO
More processing engines?
740
YES
Transmit packet to next serial processing engine
745
Transmit packet to destination
Process packet using tag information
750
755

FIG. 7

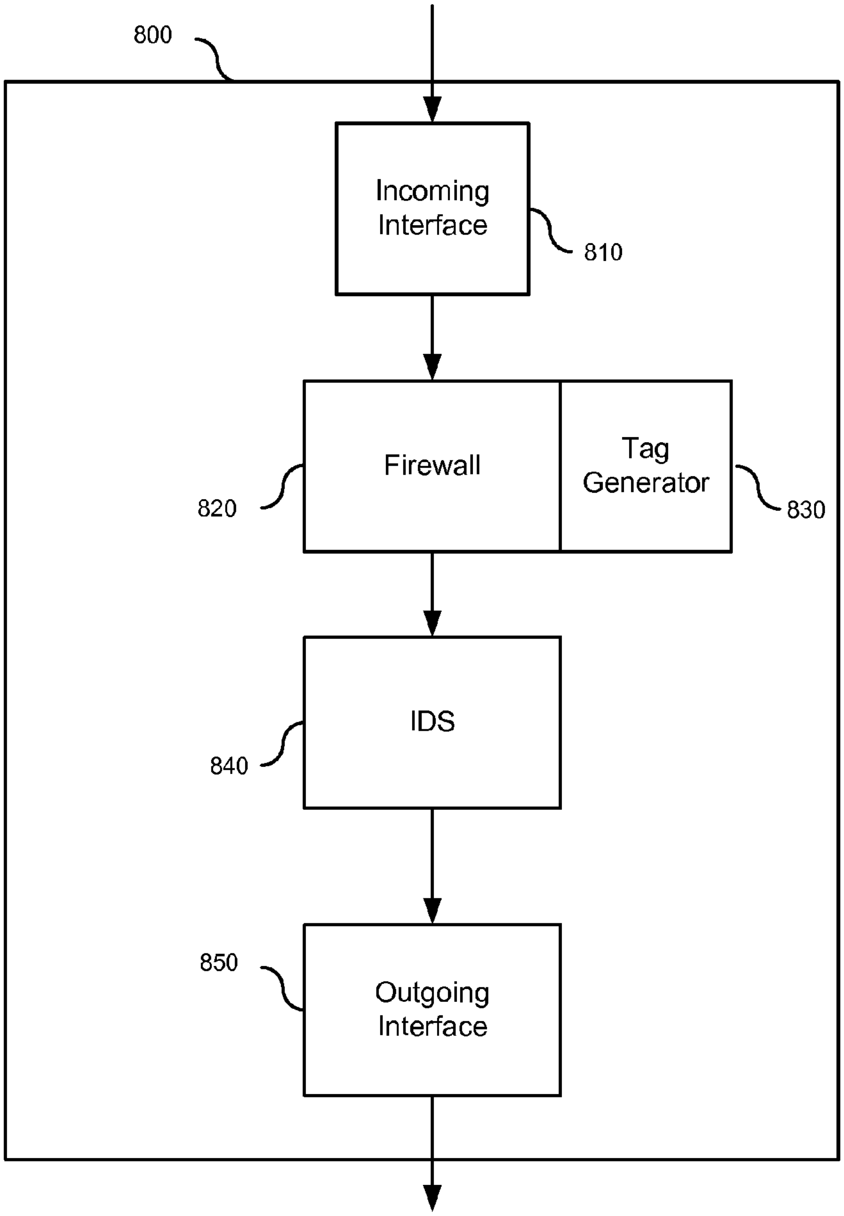

800
Incoming Interface
810
Firewall
820
Tag Generator
830
IDS
840
850
Outgoing Interface

FIG. 8

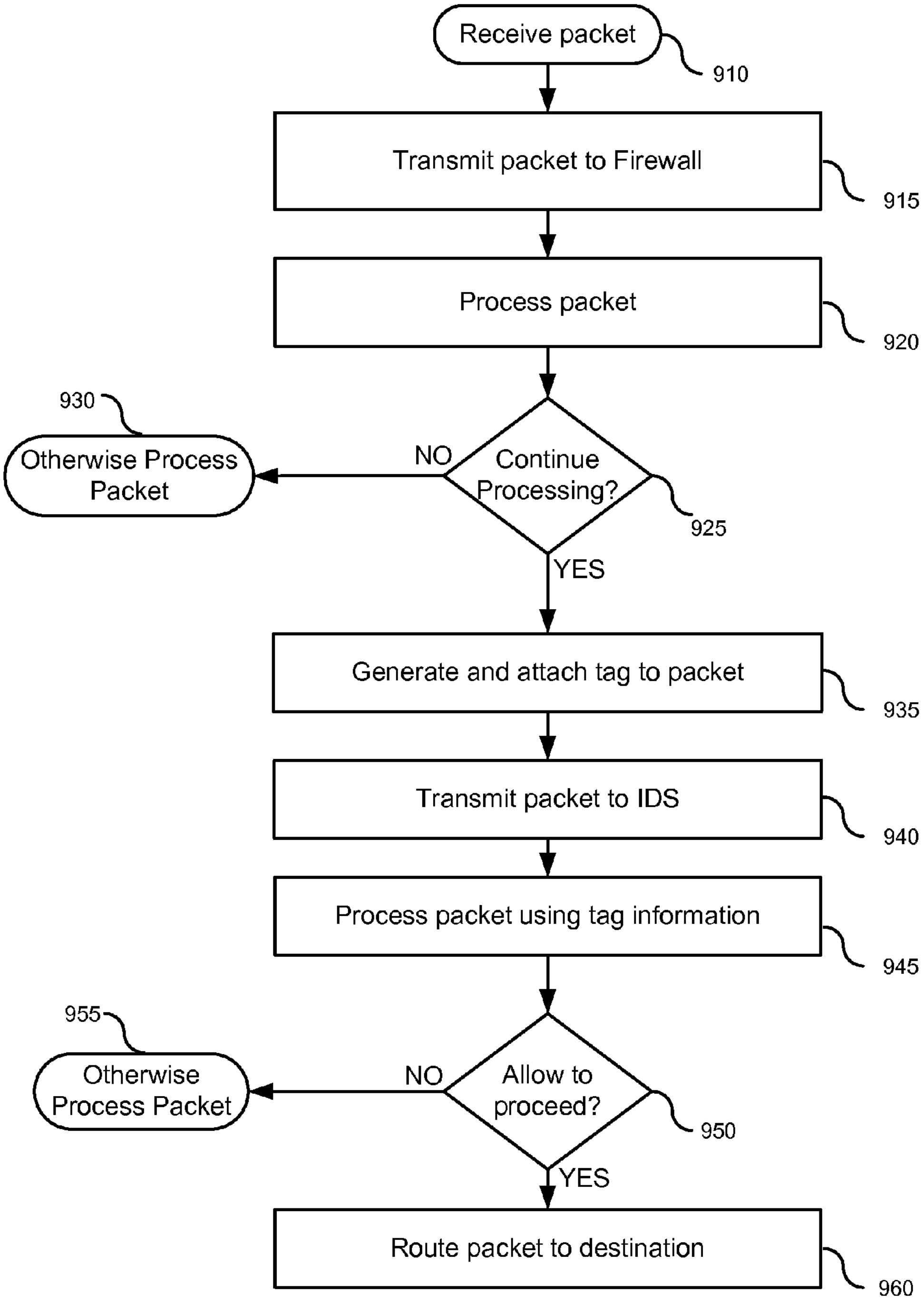

Receive packet
910
Transmit packet to Firewall
915
Process packet
920
930
Otherwise Process Packet
NO
Continue Processing?
925
YES
Generate and attach tag to packet
935
Transmit packet to IDS
940
Process packet using tag information
945
955
Otherwise Process Packet
NO
Allow to proceed?
950
YES
Route packet to destination
960

FIG. 9

PACKET PROCESSING IN A MULTIPLE PROCESSOR SYSTEM

RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 11/338,732 filed Jan. 25, 2006, which claims priority under 35 U.S.C. §119(e) based on U.S. Provisional Patent Application Ser. No. 60/704,432 filed Aug. 2, 2005 and is related to U.S. patent application Ser. No. 10/402,920, filed on Mar. 28, 2003 (now U.S. Pat. No. 7,650,634), which are herein incorporated by reference in their entirety.

FIELD OF THE INVENTION

The principles of the invention relate generally to network packet processing systems and, more particularly, to packet processing in multiple processor systems.

BACKGROUND

Network information can be split up into units of information called packets. Typically, the packets are routed dynamically between and within networks based on an efficient route between the source of the packet and the destination of the packet. Each packet typically contains a header that includes source and destination information for routing the packet and a data payload including typically a portion of the information being transmitted.

When a packet enters a network or system from an external source the packet can be examined. The examination may include determining where the packet should be routed, but also can include processing the packet to evaluate possible threats posed by the packet to the network. Different packet processing devices can be used to examine packets, for example, some packet processing devices target specific portions of a packet.

In a conventional network model, different packet processing devices can be designed to examine different layers within a packet. For example, a layered network model called the Open Systems Interconnect (OSI) model has been created by the International Standards Organization (ISO). The OSI model describes defined layers in a network operating system. Each layer has a defined input, output, and function. The OSI model defines a seven layer network and includes network and transport layers (layers 3 and 4, respectively) and an application layer (layer 7).

One type of packet processing device is a firewall. A firewall can be used to secure a network from users outside the network. The firewall checks, routes, and frequently labels all messages sent to or from users outside the network. Another packet processing device, such as an intrusion detection system, can be used to examine information being communicated with a network to recognize suspicious patterns of behavior. Information obtained by the intrusion detection system can be used to block unauthorized or disruptive users from accessing the network.

A flow-based router (FBR) allows network administrators to implement packet forwarding and routing according to network policies defined by a network administrator. FBRs allow network administrators to implement policies that selectively cause packets to be routed through specific paths in the network. FBRs can also be used to ensure that certain types of packets receive differentiated, preferential service as they are routed. Conventional routers can forward packets to their destination address based on available routing information. Instead of routing solely based on the destination

address, FBRs enable a network administrator to implement routing policies to allow or deny packets based on several other criteria including the application, the protocol, the packet size and the identity of the end system.

A packet filter can operate on the packets in the network layer, to defend a trusted network from attack by an untrusted network. Packet filters can operate at the network layer to inspect fields of the Transmission Control Protocol/Internet Protocol (TCP/IP) header including, the protocol type, the source and destination Internet Protocol (IP) address, and the source and destination port numbers.

SUMMARY

The present specification describes systems and methods for providing packet processing in a multiple processor system.

In one aspect consistent with the principles of the invention, a packet processing system is provided. The system includes a first processor for processing a packet and for creating a tag associated with the packet. The tag includes information about the processing of the packet. The system includes a second processor to receive the packet subsequent to the first processor. The second processor is configured to process the packet using the tag information.

In a second aspect consistent with the principles of the invention, a method for processing packets in a packet processing device is provided. The method includes receiving a packet at a packet processing device, directing the packet to a processor, processing the packet, creating a tag associated with the packet, where the tag includes information about the processing, forwarding the packet and the associated tag to a next processor in the packet processing device, and processing the packet at the next processor using the tag information.

In a third aspect consistent with the principles of the invention, the invention provides a packet processing system. The system includes a flow engine to route a packet among a group of processors. The system includes a first processor of the group of processors to process the packet and to create a tag to attach to the packet. The tag includes information about the processing of the packet by the first processor. The system includes a second processor of the group of processors to receive the packet from the flow engine and to process the packet including using the tag information.

In a fourth aspect consistent with the principles of the invention, a method for processing packets is provided. The method includes receiving a packet at a flow engine, routing the packet to a first processor, processing the packet at the first processor, creating and attaching a tag to the packet at the first processor, where the tag includes information about the processing useful to a next processor, transmitting the packet, including the tag, to the flow engine, routing the packet, including the tag, to the next processor, and processing the packet at the next processor using the tag information.

Implementations of the systems may include one or more of the following features. One processor of the system can be a firewall. One processor of the system can be an intrusion detection system. The tag can be appended or prepended to the packet. The tag includes data processed by the second processor. The tag information can include session information, flow information, instructions for inspection of the packet, an indication to drop the packet, or an indication to drop subsequent packets from a same session as the packet. Flows and sessions are described in U.S. patent application Ser. No. 10/072,683, filed Feb. 8, 2002, entitled "Multi-

Method Gateway-Based Network Security Systems and Methods," the contents of which are incorporated herein by reference in its entirety.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

DESCRIPTION OF DRAWINGS

FIG. **1** shows an exemplary network topology including a processing system.

FIG. **2***a* illustrates a block diagram of an exemplary processing system including multiple tag generators.

FIG. **2***b* illustrates a block diagram of an alternative processing system including a single tag generator.

FIG. **3***a* illustrates a block diagram of an alternative processing system including serial processors.

FIG. **3***b* illustrates a block diagram of an alternative processing system.

FIG. **4** illustrates an exemplary packet processing engine and a tag generator.

FIG. **5** illustrates a second exemplary packet processing engine and a tag generator.

FIG. **6** is a flowchart describing the operation of an exemplary processing system including a flow engine.

FIG. **7** is a flowchart describing the operation of an exemplary processing system including serial processors.

FIG. **8** illustrates a block diagram of an exemplary processing system including a firewall and an intrusion detection system.

FIG. **9** is a flowchart describing the operation of an exemplary processing system including a firewall and an intrusion detection system.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

FIG. **1** illustrates a network topology including a local area network (LAN) **100**, including a server **102**, several workstations (W/S) **104***a*-**104***c* (collectively, "**104**"), and a processing system **124**. Processing system **124** may include a group of processing engines **106***a*-**106***d* (collectively, "**106**") for processing packets. LAN **100** may be connected to an external network e.g., the Internet **114***b*, through processing system **124**. LAN **100** may also be connected to a second LAN **116** through a router **118**, and satellite **120**. Second LAN **116** may include a web server **110**, an email server **112**, a server **102**, several workstations **134***a*-**113***f* (collectively, "**134**") and a processing system **126**. LAN **116** may be connected to the Internet **114***a* via processing system **126**. Processing system **126** may include processing engines **128***a*-**128***b* (collectively, "**128**"). The computers, servers and other devices in the LAN may be interconnected using a number of data transmission media such as wire, fiber optics, and radio waves. Processing system **124** and processing system **126** may operate in a similar manner. Using processing system **124** by way of example, processing engines **106** may include a firewall processing engine, an intrusion detection system, a network address translation (NAT) engine and other processing engines. Processing engines **106** can act in conjunction with each other to facilitate the efficient processing of packets at different network levels. For example, a firewall processing engine can examine packet information at network layer 3 and layer 4 levels while an intrusion detection engine can look at a different network layer in the packet, such as network layer 7 information. The firewall processing engine may provide information regarding results of the firewall's processing to the intrusion detection engine in order to facilitate processing by the intrusion detection engine. The communication between processing engines is discussed in greater detail below.

FIG. **2***a* illustrates an implementation of a processing system, such as processing system **124***a*. Processing system **124***a* may include a packet interface **210** for receiving packets, a flow engine **220**, such as a flow-based router, a group of processing engines **230***a*-**230***c* (collectively, "**230**"), and a group of tag generators **240***a*-**240***c* (collectively, "**240**"). Packet interface **210** may transmit received packets to flow engine **220**, which may route the packets to processing engines **230**. Processing engines **230** may process individual packets based on some predetermined criteria. For example, processing engine **230** can be a firewall processing engine that examines packets at network layer 3 and layer 4. Processing engines **230** may be coupled to tag generators **240**. Tag generators **240** can be used to attach tags to packets following processing by processing engines **230**. Tag generators **240** can replace existing tags attached to packets with new tags. Tag generators **240** can also attach new tags to packets without altering or removing an existing tag or tags. Packets can thus have multiple attached tags. Alternatively, tag generators **240** can be used to modify an existing tag attached to packets. The tag content is described in further detail with respect to FIG. **4** below. Three processing engines **230***a*, **230***b*, and **230***c* are illustrated in FIG. **2***a* coupled parallel to each other through the flow engine **220**. Any number of processing engines **230** can, however, be included in the processing system **124***a*.

FIG. **2***b* illustrates an alternative implementation of processing system **124***a*. Processing system **124***a* may include packet interface **210** for receiving packets, flow engine **220**, such as a flow-based router, a plurality of processing engines **230***a*-**230***c* (collectively, "**230**"), and a shared tag generator **250**. Packet interface **210** may transmit received packets to flow engine **220**, which may route the packets to processing engines **230**. Processing engines **230** may process individual packets based on some predetermined criteria. For example, processing engine **230** can be a firewall processing engine that examines packets at the network layer 3 and layer 4. Processing engines **230** may be coupled to shared tag generator **250**. Shared tag generator **250** can be used to attach tags to packets following processing by processing engines **230**. Attached tags can replace existing tags or tags can be attached without modifying any existing tags. Alternatively, shared tag generator **250** can be used to modify an existing tag attached to packets. In one implementation, all processing engines **230** share one tag generator. In another implementation, a subset of processing engines **230** share one of a group of shared tag generators.

FIG. **3***a* illustrates an alternative implementation of processing system **124***a*. Processing system **124***a* may include an incoming packet interface **310** for receiving packets into processing system **124**. Received packets may pass though interface **310** to first serial processing engine **320***a*. First serial processing engine **320***a* may be coupled to a first tag generator **330***a* for attaching tags to packets after processing. First serial processing engine **320***a* may be coupled serially to second serial processing engine **320***b*. Second serial processing engine **320***b* may be coupled to second tag generator **330***b*. Additional serial processing engines can be coupled in series from second serial processing engine **320***b*. Additional tag generators can be coupled to each additional serial processing engine. Tag generators **330***a* and **330***b* (collectively, "**330**") can generate a new tag or can modify an existing tag. New or modified tags can replace existing tags or can be attached without replacing existing tags. Tags may be sent over paths **350***a* and **350***b* (collectively, "**350**") and packets may be sent over paths **305***a* and **305***b* (collectively, "**305**"). Alternatively, packets and tags may be sent over a common path. FIG. **3***a* illustrates processing system **124***a* having three serial processing engines **320***a*. Any number of processing engines **320**, however, can be included in processing system **124***a*. Interface **340** may be coupled to a last serial processing engine, in this case third serial processing engine **320***c*, through which packets exit processing system **124***a*.

Other packet processing architectures can be implemented, such as processing system **124***a* illustrated in FIG. **3***b*. FIG. **3***b* shows an alternative arrangement of processing engines and tag generators. Processing engine **370** can receive a packet from processing engines **360***a*-**360***c* (collectively, "**360**") and tag generators **365***a*-**365***b* (collectively, "**365**"). Each of the processing engines **360** may receive packets from a source, such as an interface or another processing engine (not shown). Processing engines **360** may then process packets and tag generators **365** may attach a tag to each of the packets. The packets may then be transmitted to processing engine **370**. Processing engine **370** is coupled to tag generator **375** for creating a new tag or for modifying an existing tag after processing by processing engine **370**. New or modified tags can be attached to a packet without impacting an existing tag, or a new or modified tag can replace an existing tag. After processing by processing engine **370**, packets can be transmitted to one or more additional processing engines **380***a*-**380***c* (collectively, "**380**"). Processing engines **380** may be coupled to tag generators **385***a*-**385***c* (collectively, "**385**") for attaching a tag to the packets. The packets may then be transmitted to other devices, such as processing engines for further processing of the packet or an interface for transmission outside of processing system **124***a*.

FIG. **4** illustrates a portion of the path of packet **460** in processing system **124** as illustrated in FIG. **2***a*. FIG. **5** illustrates a portion of the path of a packet in processing system **124** as illustrated in FIG. **3***a*. Referring now to FIG. **4**, packet **460** is routed by a flow engine **420** from an interface (not shown) to a first processor **410**. First processor **410** may include a processing engine **430** and a tag generator **440**. After processing, processing engine **430** may transmit a packet, which may include a header and a data payload, to tag generator **440**. Tag generator **440** may attach a tag to the packet and may transmit the packet/tag combination back to flow engine **420** for routing to a next processor **450**.

Tag generator **440** may generate tags based on the results of the processing by first processing engine **430**. The tags can include information useful to next processor **450** in processing the associated packet. Additionally, tags can include information or instructions for flow engine **420**. Tags can include information from all prior processing steps. or can include only the processing of the immediately preceding processing engine. The tag information can include, but is not limited to, some or all of the following: network layer 3 and layer 4 data, a context pointer, a cookie, a next processing context, and a communication action flag. In a network having a seven layer model, network layer 3 and layer 4 information includes information such as source IP address, destination IP address, protocol, port numbers, TCP states, running sequence numbers, and retries. The context pointer provides context information for the packet that can be useful for the next processing engine. For example, the context pointer can include session information for the packet. The session information informs the processing engine of the session the packet belongs to, can provide for consistent treatment of all packets within a particular session, and provide for expedited processing of subsequent packets belonging to a same session.

A cookie can be included that provides a composite of system-related information for the processing of a specific packet. For example, a packet subject to special inspection, resulting from a user policy configuration, can include a cookie with information about the special inspection required for use by the subsequent processing engines. The next processing context information can instruct a subsequent processing engine on actions to take following processing of the packet. The tag can also include information on which processing engine should process the packet next or can include instructions for a subsequent processing engine. For example, a firewall processing engine can attach a tag following processing that directs the intrusion detection engine to transmit the packet to a particular processing engine following processing by the intrusion detection engine. In a further example, the intrusion detection engine can be instructed to transmit the packet next to a Virtual Private Network (VPN) processing engine for encryption following intrusion detection. In one implementation, the next processing context may include both a processing engine ID, identifying the desired processing engine the instructions are directed to, as well as context information for the processing engine. The context information being provided to a VPN processing engine can include, for example, a security association for the packet. A security association can include, for example, the unique encryption keys for a session so that the VPN processing engine knows which encryption key to use on the packet and all packets in the same session.

The communication action flag can be generated to provide communication between different processing engines and between processing engines and a flow engine. For example, a communication action flag attached to a packet by a processing engine can inform the flow engine not to route any more packets from the same session to any processing engines. For example, if the intrusion detection engine determines that a packet is part of an attack, the intrusion detection engine can attach a tag to the packet instructing the flow engine to drop all incoming packets from the session upon receipt. The communication action flag can also include an indication for a processing engine or a flow engine to otherwise process the packet. Otherwise processing can include dropping, logging, alarming, and holding the packet.

Referring now to FIG. **5**, a packet may be routed by an interface **520** to a first serial processor **510**. First serial processor **510** may include a first serial processing engine **530** and a tag generator **540**. The packet may include a header and a data payload, and may be transmitted from first serial processing engine **530** to tag generator **540**. Tag generator **540** may attach a tag to the packet and may transmit a packet/tag combination to a second serial processor **550**. Tag generator **540** may generate the tag based on the results of the processing by first serial processing engine **530**. The tag can include information useful to second serial processor **550** in processing the packet. The tag information can include, but is not limited to, some or all of the following: network layer 3 and layer 4 data, a context pointer, a cookie, a next processing context, and a communication action flag. The content and use of the tag attached to the packet may be similar to the tag content and use described above with respect to FIG. **4**.

FIG. **6** provides a flowchart illustrating operations performed by processing system **124***a* of FIG. **2***a*. A packet is received by flow engine **220** from interface **210** (step **610**).

Flow engine **220** may route the received packet to first of processing engines **230** (step **620**). The processing engine processes the packet (step **630**). Processing can take numerous different forms depending on the type of processing engine. For example, a processing engine performing firewall processes can examine layer 3 and layer 4 information within a packet to search for a network attack.

After processing the packet, a determination may be made as to whether or not to continue processing the packet (step **640**). For example, if the processing engine is a firewall, processing can determine that the packet is part of an attack. As a result, the processing engine can otherwise process (e.g., drop, log, alarm, or hold) the packet (step **650**). If the processing is to continue, a tag may be attached to the packet by the tag generator associated with the processing engine (step **660**). For example, if the packet is suspected of an attack based on the analysis by a firewall processing engine, the tag can include instructions for a subsequent one of processing engines **230** (for example, an intrusion detection engine) to make a careful investigation of the packet to determine whether or not the packet is an attack, and to drop the packet if it is an attack. In an alternative implementation, instead of dropping the attack packet, a tag may be attached to the packet that includes a communication action flag for flow engine **220**. The communication action flag may instruct flow engine **220** to drop the packet and any received packets matching the session of the packet. Conversely, if the packet is determined by the firewall processing engine to be a packet that is not suspect, the tag can include information informing the intrusion detection engine that no detailed investigation is necessary. The tag can also include instructions for one of processing engines **230** or flow engine **220** to "otherwise process" the packet. "Otherwise processing" the packet can include, for example, dropping, logging, alarming, holding, and alerting, each of which may result in the content of the packet being modified.

The packet with the attached tag may be transmitted back to flow engine **220** for routing to subsequent processing engine **230** (step **670**), as shown in FIG. **4**. Flow engine **220** may then determine if more processing engines **230** are used to process the packet prior to routing the packet to a destination outside processing system **124***a* (step **680**). Flow engine **220** can use information in the tag to determine what further processing is required. If no other processing engines **230** are to process the packet, flow engine **220** may route the packet through interface **210** to the destination (step **695**). If other processing engines **230** are to process the packet, flow engine **220** may route the packet to the next of processing engines **230** to process the packet (step **685**). Additionally, flow engine **220** can receive instructions within the tag from one of processing engines **230**. For example, if the firewall processing engine determines that a packet is part of an attack, a tag including a communication action flag can be sent to flow engine **220** informing flow engine **220** not to route any more packets from the same session as the packet.

A next one of processing engines **230** may then process the packet using information obtained from the attached tag (step **690**). The next one of processing engine **230** may examine the tag for information based on previous processing of the packet. The tag can provide information leading to expedited or more intensive processing by the next one of processing engines **230**. For example, the tag can provide information indicating that a previous one of processing engines **230** determined that the packet was possibly part of an attack and requires detailed examination by a next one of processing engines **230**. Alternatively, the tag can provide information indicating that a previous one of processing engines **230** determined that the packet was not a threat and does not require detailed examination by a subsequent one of processing engines **230**.

After processing, flow may return to step **640** to determine if processing of the packet should continue. For example, the packet can be cleared by the firewall processing engine only to be recognized by the intrusion detection engine as part of an attack. Upon discovery of the attack, the intrusion detection engine can determine that no further packets from the attack session should be processed. If the processing is terminated, the packet can be otherwise processed, for example by dropping the packet (step **650**). If the processing is allowed to continue, a new tag may be attached to the packet reflecting the processing (step **660**). The new tag can be attached to the packet by a tag generator for a particular one of processing engines **230**, or the new tag can be attached to the packet by a shared tag generator **250** (FIG. **2***b*). Alternatively, new data may be attached to the old tag instead of attaching a new tag to the packet.

In another implementation, a packet to be dropped can have a tag attached including instructions to flow engine **220** not to route any further packets from the session of the packet. The process from step **640** to step **690** may be repeated for each of processing engines **230** until no other processing engines **230** are to process the packet. For example, flow engine **220** can determine, based on prior processing steps and tag information, if further processing is required. When no other processing engines **230** are to process the packet, flow engine **220** may route the packet to the destination (step **695**). In one implementation, the tag may be discarded by flow engine **230** before routing the packet outside processing system **124***a*.

FIG. **7** shows a flowchart illustrating processing system **124***a* of FIG. **3***a*. Processing system **124***a* may receive a packet at interface **310** (step **710**). Interface **310** may transmit the packet to first serial processing engine **320***a* (step **715**). First serial processing engine **320***a* may process the packet (step **720**). As described above with respect to FIG. **6**, processing can include examination of different network layers within the packet. First serial processing engine **320***a* can be a firewall as discussed above with respect to FIG. **6**, or some other processing engine. After processing, first serial processing engine **320***a* may determine if the packet should be transmitted to a next one of serial processing engines **320** (e.g., second serial processing engine **320***b*) for further processing (step **725**).

If first serial processing engine **320***a* determines that the packet should not be processed further, the packet may be otherwise processed (step **730**). If first serial processing engine **320***a* determines that the packet can continue processing, a tag may be attached to the packet by tag generator **330** (step **735**). If there are more serial processing engines **320** (step **740**) then the packet may be transmitted to next of serial processing engines **320** (e.g., second serial processing engine **320***b*) (step **745**). The next one of serial processing engines **320** may then process the packet using the tag information contained in the tag (step **750**). After the next one of serial processing engines **320** processes the packet using the tag information, the flow may return to step **725** for a determination of whether or not to continue processing the packet. If the packet is not to be processed further, the packet may be otherwise processed (step **730**). If the processing is to continue, a new tag may be attached to the packet by one of tag generators **330** (step **735**). In an alternative implementation, new data may be attached to an existing tag. The tag information can include information from all prior processing steps or can include only the processing of the immediately preceding one of processing engines **320**. The tag can also include instructions for processing engines **320** to otherwise process the packet. Otherwise processing the packet can include dropping, logging, alarming, and holding. The tag information can include the same information as described above with respect to FIG. **4**.

The flow from step **725** to step **750** continues for each subsequent one of serial processing engines **320** (e.g., third serial processing engine **320***c*) until no further serial processing engines **320** remain to process the packet. The last of serial processing engines **320** (e.g., third serial processing engine **320***c* in FIG. **3**) may transmit the packet to the destination as defined, for example, by the destination IP address of the packet (step **755**). In one implementation, a tag may be removed by last serial processing engine **320** before transmission of the packet to interface **340**. In another implementation, the tag may be removed by interface **340** prior to transmission of the packet to the destination.

FIG. **8** illustrates a processing system **800** that may include an incoming interface **810** that receives a packet from an external source, such as an external network. Incoming interface **810** may route the packet to a firewall **820** for processing. Firewall **820** includes a tag generator **830** for attaching a tag to a packet. A packet/tag combination can be transmitted from firewall **820** to an IDS **840** for further processing. After processing, the packet can be transmitted to an outgoing interface **850**. Outgoing interface **850** may then route the packet out of processing system **800** to the packet's destination.

FIG. **9** shows a flowchart illustrating exemplary processing of processing system **800** of FIG. **8**. Processing system **800** may receive a packet at interface **810** (step **910**). Interface **810** may transmit the packet to firewall **820** (step **915**). Firewall **820** may process the packet (step **920**). The processing by firewall **820** may include examination of the packet at network layer 3 and layer 4. Processing by firewall **820** may also include performing a session look-up on the packet in order to determine to which session the packet belongs. Based on the examination, firewall **820** may determine whether to transmit the packet to the next processor (step **925**). If examination by firewall **820** reveals that the packet is a threat, the packet can be otherwise processed (e.g., dropped) (step **930**). If the packet is allowed to proceed, tag generator **830** may generate and attach a tag to the packet (step **935**). The tag may include session ID information for the packet as determined by firewall **820**. The tag can also include log information for the packet.

The firewall may then transmit a packet/tag combination to IDS **840** for processing (step **940**). IDS **840** may use the session ID contained within the tag so that the IDS does not have to perform a session look-up on the packet. As a result, IDS **840** can apply policies on how to process the packet based on the session ID without performing a session lookup (step **945**). After processing, IDS **840** may determine whether or not the packet is allowed to proceed out of processing system **800** (step **950**). If, based on the processing, the packet is not allowed to proceed, the packet can be otherwise processed (e.g., dropped) (step **955**). If the packet is allowed to proceed, the tag may be removed and the packet may be routed through interface **850** to the packet's destination (step **960**).

Tags can be appended or prepended to the packet. A new tag can be generated with each processing or the tag can have new data appended or prepended to the existing tag. The tag can include instructions as well as data to be processed by the processing engines or by the flow engines.

In one implementation, a tag generator can be used to attach a tag prior to a determination of continued processing. A tag can be automatically generated following processing by a processing engine and then a determination can be made as to whether or not the processing should continue, or the packet with tag can be forwarded to another device for a determination of continued processing. In another implementation, the processing engines may include a tag analyzer for analyzing the tag information in order to determine what level of processing is required for the packet.

In one implementation, each tag generator and processing engine pair can be integrated on one printed circuit board ("PCB") or alternatively on one integrated circuit ("IC"). In another implementation, multiple pairs of processors and tag generators can be integrated on one PCB or on one IC. In a further implementation, a plurality of processing engines can share the same tag generator. Additionally, in an implementation including a flow engine, the flow engine can be integrated with a plurality of processing engines on one PCB or on one IC. Further, the flow engine can be integrated with a plurality of processing engines and at least one tag generator on one PCB or one IC.

The invention and all of the functional operations described herein can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Method steps of the invention can be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method steps can also be performed by, and apparatus of the invention can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

To provide for interaction with a user, the invention can be implemented on a computer having a display device, e.g., a

CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

A number of implementations of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other implementations are within the scope of the following claims.

What is claimed is:

1. A system comprising:

a first engine to:

route a packet to a second engine, and

route the packet to a third engine, after receiving the packet from the second engine;

the second engine to:

process the packet, and

associate a tag with the packet, the tag including information about the processing of the packet; and

the third engine to:

process the packet using the information included in the tag,

where the second engine and the third engine comprise a firewall processing engine, an intrusion detection system, or a network address translation (NAT) engine,

where the second engine is different than the third engine, and

where the second engine and the third engine are included on one integrated circuit.

2. The system of claim **1**, where the one integrated circuit includes a central processing unit.

3. The system of claim **1**, where the first engine is further to:

determine routing of the packet using the information included in the tag.

4. The system of claim **3**, where the tag includes information associated with a next engine that is to process the packet, and

where the first engine is further to route the packet to the next engine based on the information associated with the next engine.

5. The system of claim **4**, where the information, associated with the next engine, includes identification information for the next engine.

6. The system of claim **1**, where the first engine includes a flow-based router.

7. The system of claim **1**, where the packet includes a first packet and the tag includes a first tag,

where the second engine is to process a second packet, and

where the third engine is to:

process the second packet, and

associate a second tag with the second packet, the second tag including information associated with processing of the second packet by the third engine, or including information associated with processing of the second packet by the second engine and processing of the second packet by the third engine.

8. The system of claim **1**, where the third engine is further to:

add data to the tag, the data including information, for at least one subsequent engine, associated with processing of the packet by the third engine.

9. A method comprising:

routing, using a first engine, a packet to a second engine that is different than the first engine;

processing, using the second engine, the packet;

associating, using the second engine, a tag with the packet, where the tag includes information associated with the processing of the packet using the second engine;

routing, using the first engine and based on the information included in the tag, the packet to a third engine, after receiving the packet from the second engine,

where the third engine is different than the first engine and the second engine; and

processing, using the third engine and based on the information included in the tag, the packet,

where the second engine and the third engine include a firewall processing engine, an intrusion detection system, or a network address translation (NAT) engine.

10. The method of claim **9**, where the information, included in the tag, includes identification information for another engine to process the packet after the packet has been processed using the third engine,

the method further comprising:

routing the packet to the other engine based on the identification information for the other engine.

11. The method of claim **9**, where the second engine and the third engine are included on one integrated circuit, and where the one integrated circuit includes a central processing unit.

12. The method of claim **9**, where the tag further includes information associated with a fourth engine that is to process the packet, and

the method further comprising:

routing the packet to the fourth engine using the information associated with the fourth engine.

13. The method of claim **9**, further comprising:

removing the tag; and

routing the packet to a destination of the packet when the tag is removed.

14. The method of claim **9**, where the packet corresponds to a first packet and the tag corresponds to a first tag, and

where the method further comprises:

processing a second packet using the second engine;

processing the second packet using the third engine; and

associating a second tag with the second packet, the second tag including information associated with processing of the second packet using the second engine and processing of the second packet using the third engine.

15. The method of claim **9**, further comprising:

adding data to the tag, the data including information associated with processing of the packet using the third engine.

16. The method of claim **9**, further comprising:

determining, prior to associating the tag with the packet, whether the packet is allowed to proceed to the third engine; and

generating the tag when the packet is allowed to proceed to the third engine.

17. A device comprising:

a first engine to:

route a packet to a second engine that is different than the first engine, and

route the packet to a third engine, after receiving the packet from the second engine, where the third engine is different than the first engine and the second engine;

the second engine to:

process the packet, and

associate a tag with the packet,

where the tag includes information associated with the processing of the packet; and

the third engine to:

process the packet using the information included in the tag,

where the second engine and the third engine comprise a firewall processing engine, an intrusion detection system, or a network address translation (NAT) engine, and

where the second engine and the third engine are included on one integrated circuit.

18. The device of claim **17**, where the one integrated circuit includes a central processing unit.

19. The device of claim **17**, where the second engine is further to:

generate the tag when the packet is allowed to proceed to the third engine.

20. The device of claim **17**, where the tag includes identification information associated with another engine that is to process the packet after the packet has been processed using the third engine, and

where the first engine is to route the packet to the other engine based on the identification information associated with the other engine.

* * * * *

EXHIBIT B

US007779459B2

(12) United States Patent
Mao et al.

(10) Patent No.: US 7,779,459 B2
(45) Date of Patent: *Aug. 17, 2010

(54) **METHOD AND APPARATUS FOR IMPLEMENTING A LAYER 3/LAYER 7 FIREWALL IN AN L2 DEVICE**

(75) Inventors: **Yu Ming Mao**, Milpitas, CA (US); **Roger Jia-Jyi Lian**, San Jose, CA (US); **Guangsong Huang**, Sunnyvale, CA (US); **Lee Chik Cheung**, San Jose, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 343 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/869,287**

(22) Filed: **Oct. 9, 2007**

(65) **Prior Publication Data**

US 2008/0034414 A1 Feb. 7, 2008

Related U.S. Application Data

(63) Continuation of application No. 09/967,878, filed on Sep. 28, 2001, now Pat. No. 7,302,700.

(51) **Int. Cl.**
G06F 17/00 (2006.01)
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **726/11**; 713/150

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,544,322	A	8/1996	Cheng et al.
5,708,654	A	1/1998	Arndt et al.
5,889,953	A	3/1999	Thebaut et al.
5,905,859	A	5/1999	Holloway et al.
5,918,018	A	6/1999	Gooderum et al.
5,968,176	A	10/1999	Nessett et al.
6,000,045	A	12/1999	Lewis
6,041,058	A	3/2000	Flanders et al.
6,115,472	A	9/2000	Shimizu et al.
6,131,120	A	10/2000	Reid
6,141,755	A	10/2000	Dowd et al.
6,182,226	B1	1/2001	Reid et al.
6,212,558	B1	4/2001	Antur et al.
6,219,707	B1	4/2001	Gooderum et al.
6,233,688	B1	5/2001	Montenegro
6,304,973	B1	10/2001	Williams

(Continued)

FOREIGN PATENT DOCUMENTS

JP	06-311161	11/1994

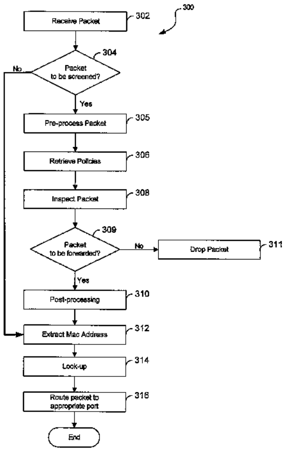
(Continued)

Primary Examiner—Christian LaForgia
(74) *Attorney, Agent, or Firm*—Harrity & Harrity, LLP

(57) **ABSTRACT**

Methods and apparatus for transferring packets in a packet switched communication system. A system is provided that includes an L2 device including a controller determining for each packet received whether the received packet is to be inspected, an inspection device operable to inspect and filter packets identified by the controller including using a zone specific policy and an L2 controller for transferring inspected packets in accordance with L2 header information using L2 protocols.

23 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

6,684,253	B1	1/2004	Whitaker et al.	
6,754,716	B1	6/2004	Sharma et al.	
6,763,469	B1 *	7/2004	Daniely	726/11
6,961,771	B2	11/2005	Sato	
7,047,561	B1	5/2006	Lee	
7,103,055	B2	9/2006	Kadambi et al.	
7,302,700	B2 *	11/2007	Mao et al.	726/11
2001/0042213	A1	11/2001	Jemes et al.	
2002/0053020	A1 *	5/2002	Teijido et al.	713/153

FOREIGN PATENT DOCUMENTS

WO	WO 95/35610	12/1995

* cited by examiner

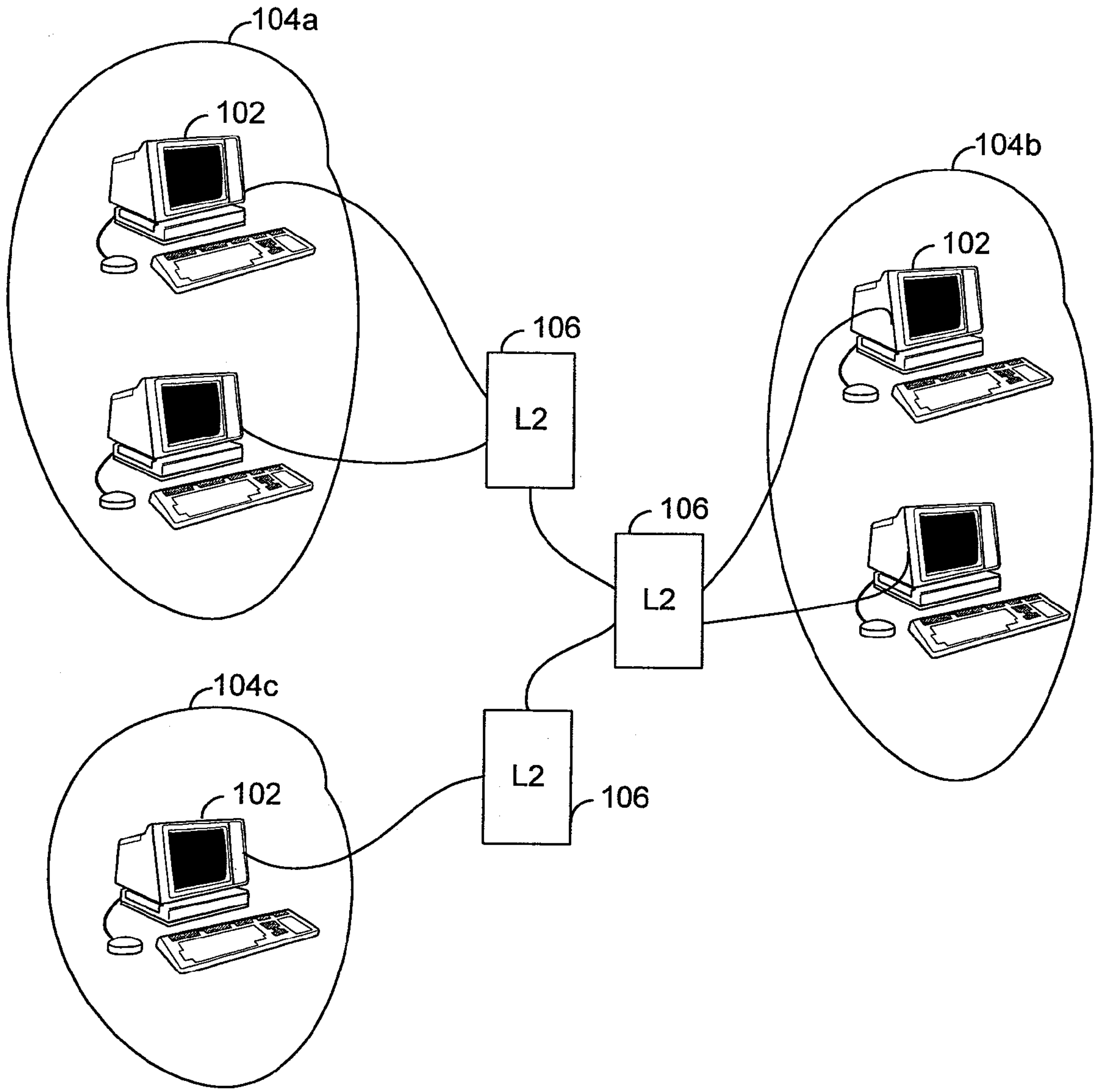


Fig. 1

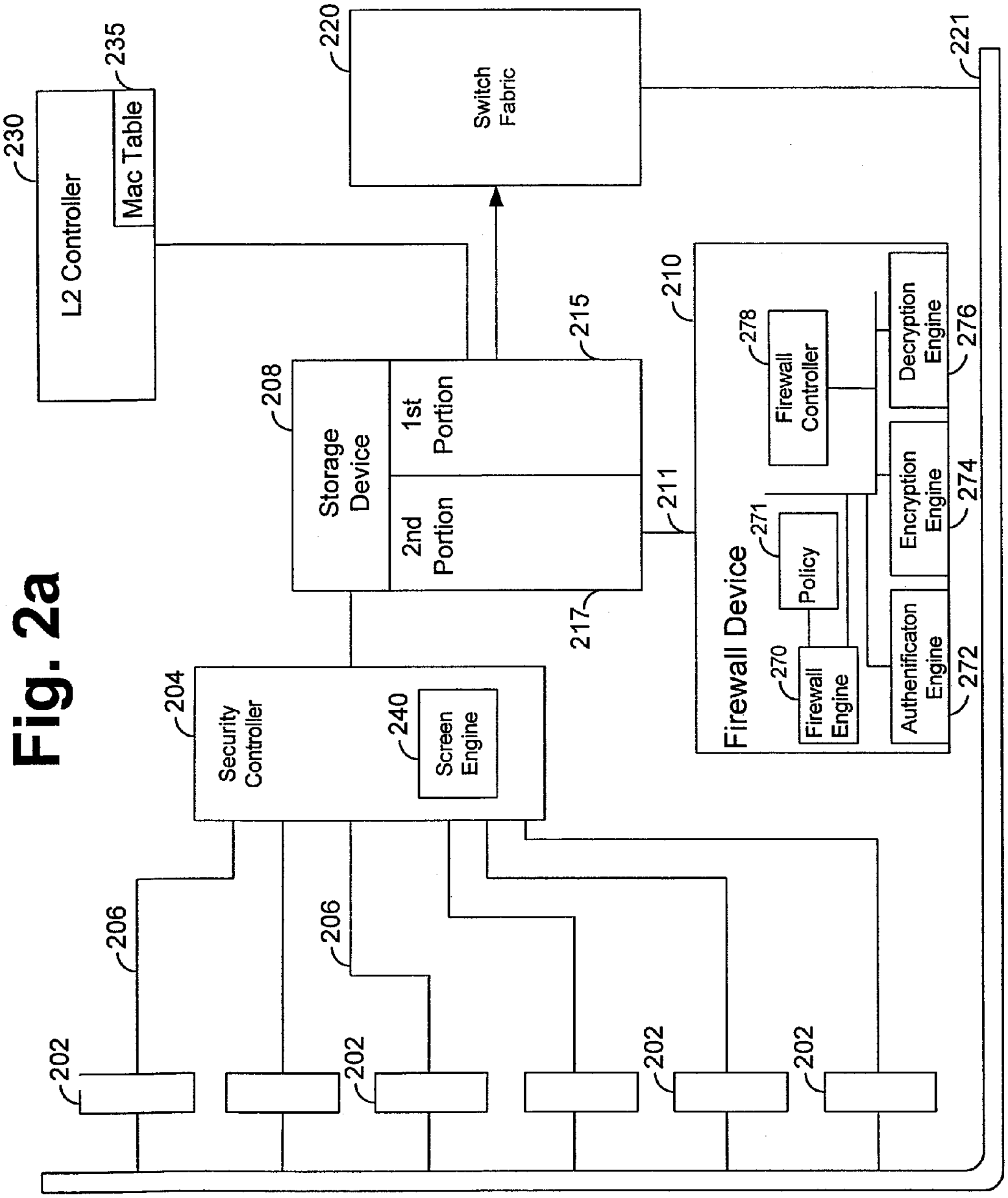


Fig. 2a

230
L2 Controller
235
Mac Table
220
Switch Fabric
221
208
Storage Device
1st Portion
2nd Portion
215
217
211
210
Firewall Device
278
Firewall Controller
271
Policy
270
Firewall Engine
Authentificaton Engine
272
Encryption Engine
274
Decryption Engine
276
204
Security Controller
240
Screen Engine
206
206
202
202
202
202

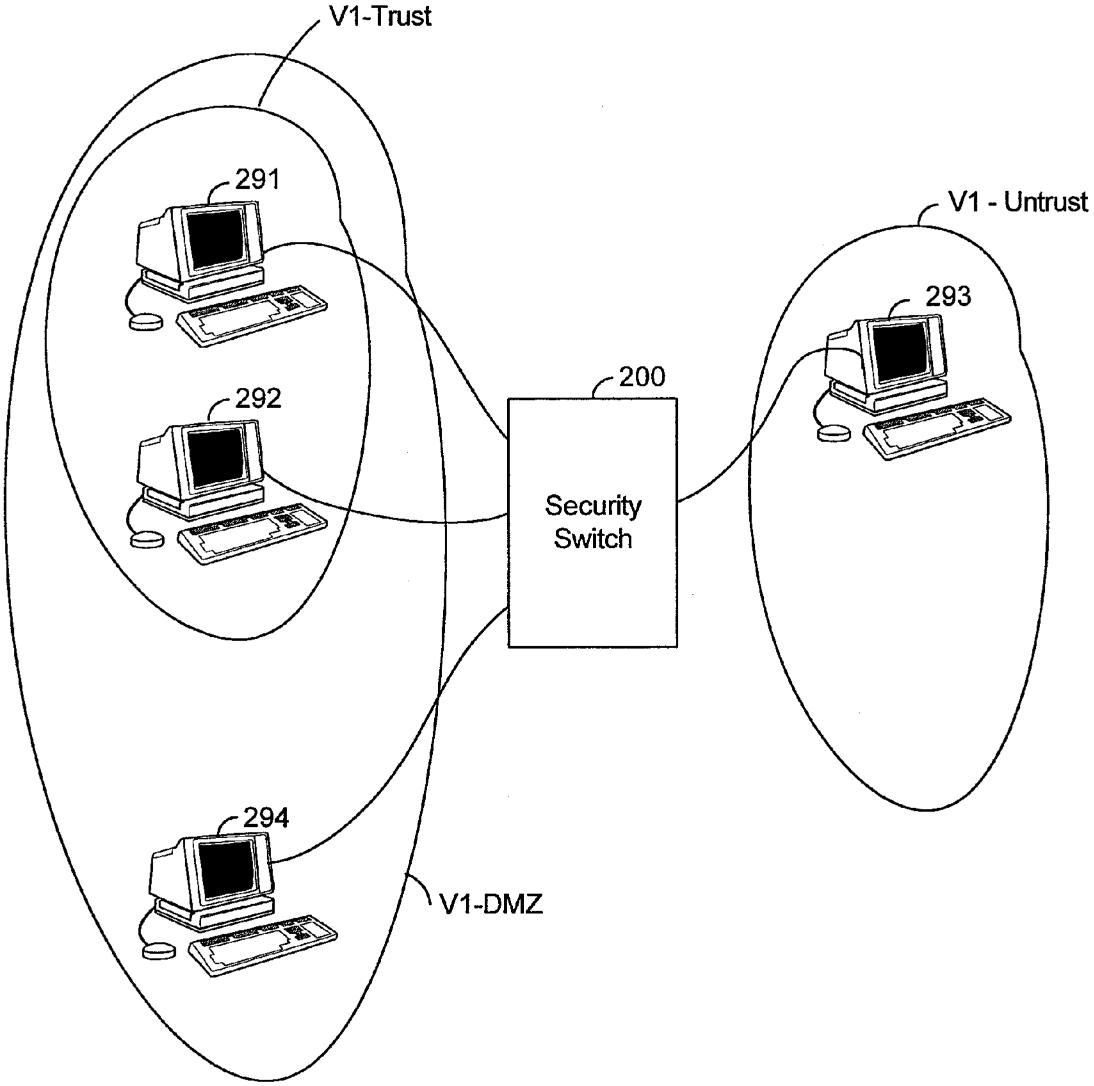

V1-Trust
291
292
V1 - Untrust
293
200
Security Switch
294
V1-DMZ

Fig. 2b

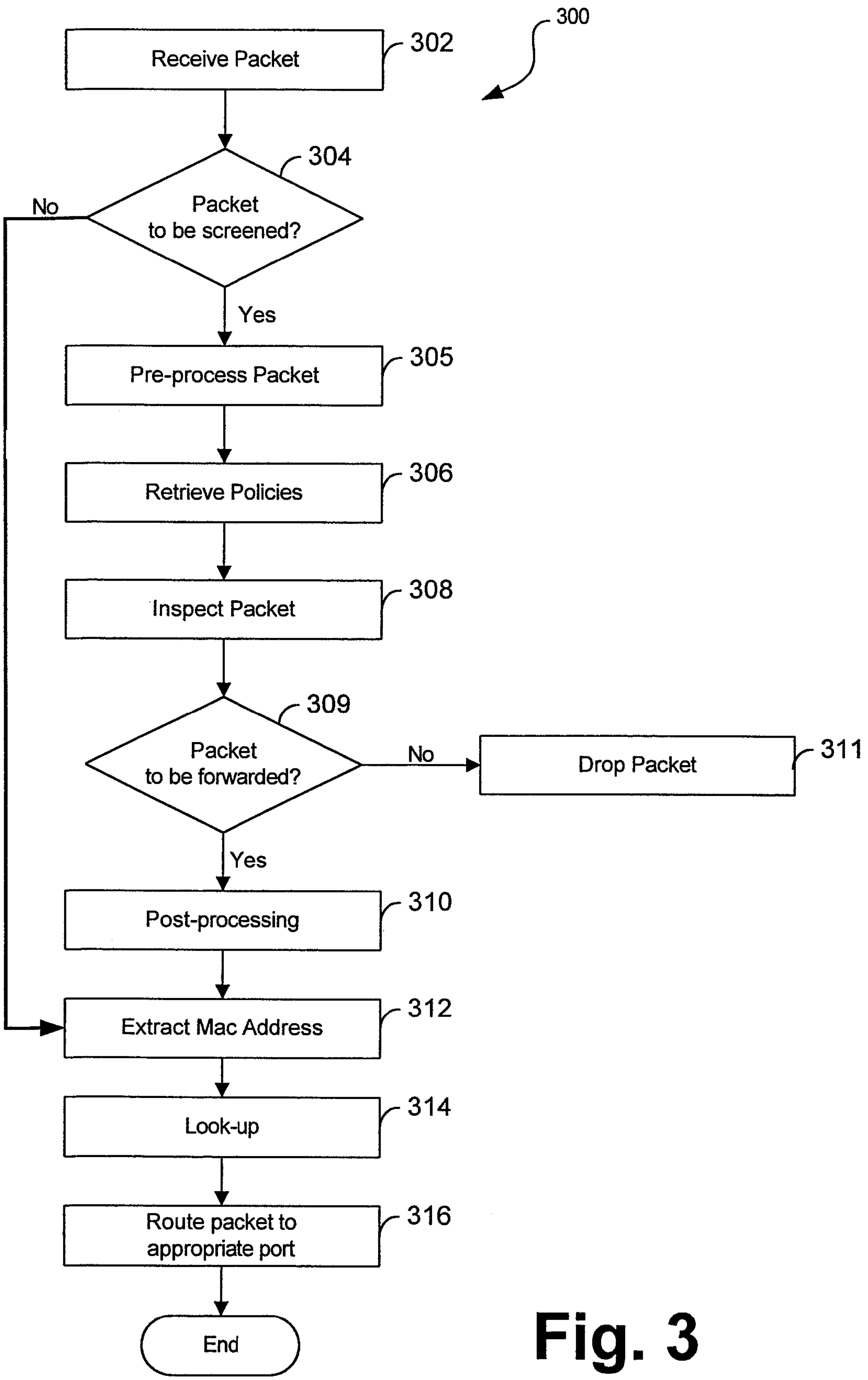

300
Receive Packet
302
Packet to be screened?
304
No
Yes
Pre-process Packet
305
Retrieve Policies
306
Inspect Packet
308
Packet to be forwarded?
309
No
Drop Packet
311
Yes
Post-processing
310
Extract Mac Address
312
Look-up
314
Route packet to appropriate port
316
End

Fig. 3

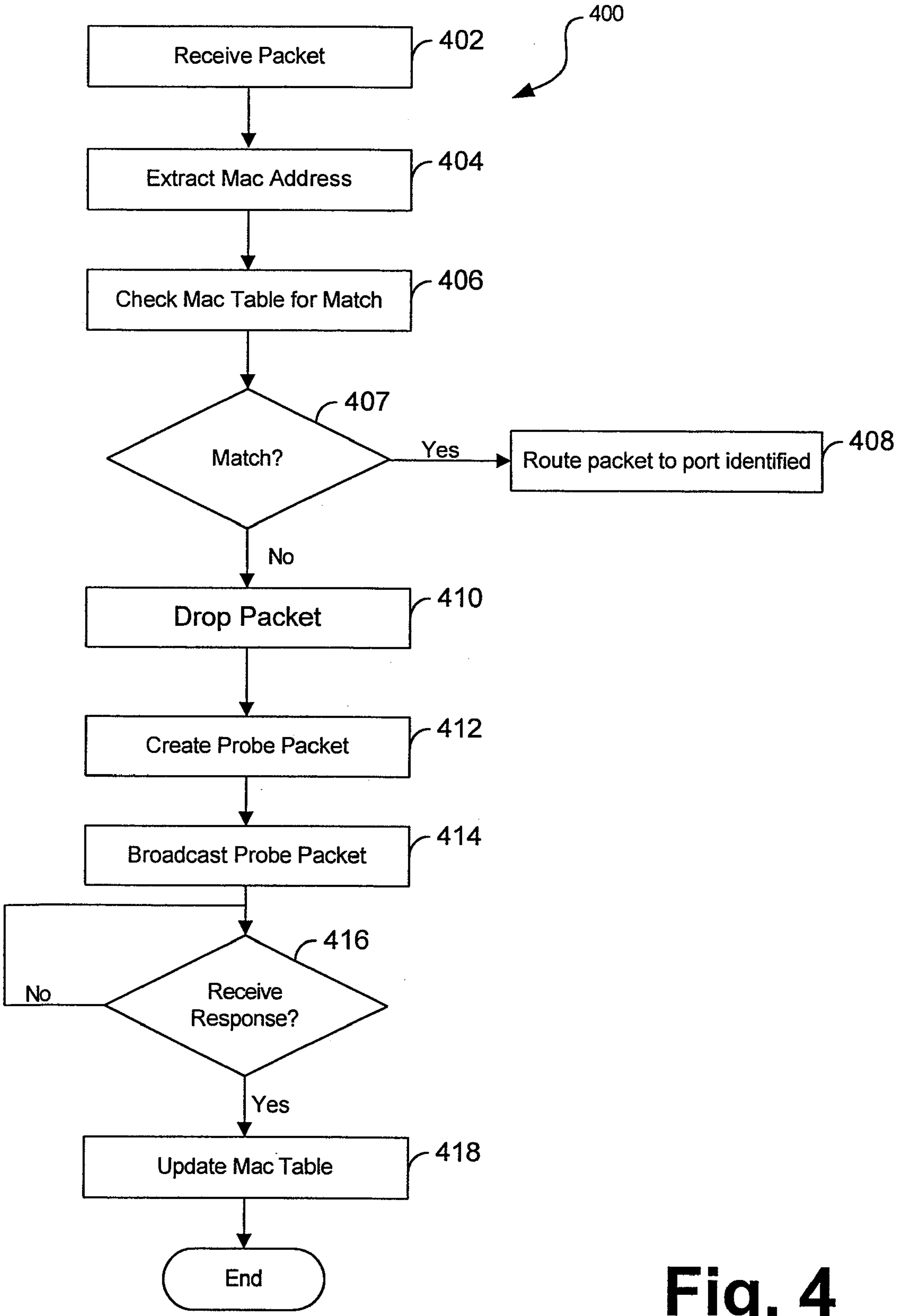

400
Receive Packet
402
Extract Mac Address
404
Check Mac Table for Match
406
Match?
407
Yes
Route packet to port identified
408
No
Drop Packet
410
Create Probe Packet
412
Broadcast Probe Packet
414
No
Receive Response?
416
Yes
Update Mac Table
418
End

Fig. 4

METHOD AND APPARATUS FOR IMPLEMENTING A LAYER 3/LAYER 7 FIREWALL IN AN L2 DEVICE

RELATED APPLICATION

This application is a continuation of U.S. patent application Ser. No. 09/967,878 filed Sep. 28, 2001, the disclosure of which is incorporated herein by reference.

BACKGROUND

The present invention relates generally to data routing systems, and more particularly to methods and apparatus for providing secure communications on a network.

A packet switch communication system includes a network of one or more switches or routers connecting a plurality of users. A packet is the fundamental unit of transfer in the packet switch communication system. A user can be an individual user terminal or another network.

A layer 2 (L2) switch is a switching device which receives packets containing data or control information on one port, and based on a media access connection (MAC) address contained within the packet, switches the packet out another port. Conventional L2 switches perform this switching function by evaluating layer 2 (L2) header information contained within the packet in order to determine the proper output port for a particular packet. The L2 switch includes a table that maps MAC addresses with output ports. If a MAC address is unknown (i.e., there is no corresponding entry in the table), then the corresponding packet is broadcast to all output ports with the hope that another component in the packet switched communication system will recognize the MAC address (and pass back information to the forwarding L2 switch to update its table). Other types of L2 devices include bridges.

A router is a switching device which receives packets containing data or control information on one port, and based on destination information contained within the packet, routes the packet to a next hop to/toward the destination. Conventional routers perform this switching function by evaluating layer 3 (L3) header information contained within the packet in order to determine a next hop for a particular packet. The layer 3 information includes an IP address associated with the intended destination (as well as source address) for the packet.

The network coupling the users can be an intranet, that is, a network connecting one or more private servers such as a local area network (LAN). Alternatively, the network can be a public network, such as the Internet, in which data packets are passed over untrusted communication links. The network configuration can include a combination of public and private networks. For example, two or more LAN's with individual terminals can be coupled together using a public network such as the Internet. Data security issues can arise when public and private networks are linked or when distinct networks are coupled. For example, conventional packet switched communication systems that include links between public and private networks typically include security measures for assuring network access control and data integrity.

In order to assure individual packet security, packet switched communication systems can include encryption/decryption services. Prior to leaving a trusted network (or portion of a network), individual packets can be encrypted to minimize the possibility of data loss while the packet is transferred over an untrusted (e.g., public) network (or portion thereof). Upon receipt at a destination or another trusted portion of the communication system (e.g., at a firewall just

before the destination), the packet can be decrypted and subsequently delivered to its intended destination. The use of encryption and decryption allows for the creation of a virtual private network (VPN) between users separated by untrusted communication links.

In addition to security concerns for the data transferred over the public portion of the communications system, the private portions of the network must safeguard against intrusions through the gateway provided at the interface of the private and the public networks. A firewall is a device that can be coupled in-line between a public network and private network for screening packets received from the public network. A firewall is a particular type of L3/L4 device that can be used to enforce policy and filtering functions. A firewall can include one or more engines for inspecting, filtering, authenticating, encrypting, decrypting and otherwise manipulating received packets. Conventional firewalls use L3 and L4 header information including IP addresses associated with the source and destination of a given packet being processed. Received packets are inspected and thereafter forwarded or dropped in accordance with the policies associated with the given domain.

SUMMARY

In one aspect, the invention provides an L2 device in a packet switched communication system. The packet switched communication system has plural zones and each zone represents a distinct security domain and has an associated policy for use in inspecting packets entering/exiting an associated zone. The L2 device includes at least one port coupled to a terminal unit included in a first security zone, at least one port coupled to a terminal unit included in a second security zone, a controller determining for each packet received whether the received packet is destined for another zone, a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy and an L2 switching engine. The L2 switching engine is operable to immediately route to a port all intra-zone packets passing through the L2 device using a table of MAC addresses and corresponding ports, and only route to a port inter-zone packets that are retained after the inspection by the firewall engine.

In another aspect, the invention provides an L2 device in a packet switched communication system. The L2 device includes a controller determining for each packet received whether the received packet is to be transferred intra-zone or inter-zone, a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy and an L2 switching engine operable to immediately route to a port all intra-zone packets passing through the L2 device using a table of MAC addresses and corresponding ports and only route to a port inter-zone packets that are retained after the inspection by the firewall engine.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be transferred inter-zone and a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be transferred inter-zone and an inspection device operable to inspect and filter inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be inspected, an inspection device operable to inspect and filter packets identified by the controller including using a zone specific policy and an L2 controller for transferring inspected packets in accordance with L2 header information using L2 protocols.

Aspects of the invention can include one or more of the following features. The inspection device can be a firewall including a layer 3 firewall device, a layer 4 firewall device and a layer 7 firewall device. The inspection device can be a firewall that filters based on layer information other than layer 2 header information. The controller can determine each packet that is to pass between security zones and the inspection device only processes inter-zone traffic. The controller can determine each packet that is to remain in a single security zone and the inspection device immediately routes intra-zone packets. The device can route traffic using the MAC address in the layer 2 header of a given packet to determine an egress port on the device to which the packet is to be routed.

The device can include a storage element for storing packets that are to be inspected and an L2 controller for transferring packets through the device including determining an egress port for transferring a given packet using a destination MAC address in the given packet and a MAC address table that includes a mapping of MAC addresses and associated egress nodes.

The memory element can include a first and second portion. The first portion can store packets to be transferred through the device and the second portion can store packets waiting for inspection. The device can be a L2 switch or an L2 bridge.

In another aspect, the invention provides a method for transferring packets in a communication network including receiving a packet at an L2 device, determining whether the received packet is to be transferred inter-zone and inspecting and filtering inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides a method for transferring packets in a communication network including receiving a packet at an L2 device, determining whether the received packet is to be inspected and inspecting and filtering identified packets using a zone specific policy prior to transferring the packet through the L2 device using L2 protocols.

In another aspect, the invention provides a method for switching packets in a communication network including receiving a packet at an interface of an L2 device, determining if a destination MAC address associated with the received packet is known and, if not, holding the received packet a predetermined amount of time without transferring the packet to any port of the L2 device, creating a probe packet that includes the unknown MAC address and broadcasting the probe packet to all interfaces except the receiving interface.

Aspects of the invention can include one or more of the following features. The probe packet can include a time to life (TTL) field in a IP header and the method can include setting a value of the TTL field such that a downstream node having the unknown MAC address and receiving the probe cell will return an expired message to the L2 device. The method can include dropping the packet after the expiration of the predetermined amount of time. The packet can be dropped if the MAC address is unknown. The method can include receiving a response from on one of the broadcast interfaces and updating a table indicating a previously unknown MAC address is associated with the responding interface.

In another aspect, the invention provides method of providing secure communications between users without requiring encryption and decryption services at a respective user.

The method includes identifying first and second users, coupling the first and second users through two or more L2 devices over a communication network and specifying a virtual private network for communications between the first and second users. The virtual private network is defined between a first and second L2 device in the network. The method includes receiving a packet at either the first or the second L2 device, determining whether the received packet is associated with the virtual private network and encrypting and decrypting as appropriate identified packets using local encryption and decryption services prior to transferring the packet through the L2 device using L2 protocols.

Aspects of the invention can include one or more of the following features. The step of determining can include using a destination MAC address associated with the packet to identify a virtual private network.

In another aspect, the invention provides a virtual private network for providing secure communications between users without requiring encryption and decryption services at a respective user. The virtual private network includes first and second L2 devices coupling first and second users over a communication network where each of the first and second L2 devices includes a screening mechanism determining whether a received packet is associated with the virtual private network and encryption and decryption services operating on packets associated with the virtual private network prior to a transfer of the packet through the L2 device using L2 protocols.

Aspects of the invention can include one or more of the following advantages. A packet switched communication system is provided that allows for the creation of plural security zones within a single device without requiring changes to the network or terminal configuration. Between each zone, a terminal unit can communicate with other terminal units without the knowledge of, yet receiving the benefits of, L2 switching and up to layer 7 security filtering as discussed below. A packet switched communication system is provided that includes L2 switch and firewall functionality. The packet switched communication system acts as an IEEE 802.1Q VLAN L2 conventional switch forwarding/filtering based on MAC-address for all intra-zone communications. The packet switched communication system allows L2 switching among multiple ports inside a given security zone. The L2 switch also provides up to layer 7 security firewall protections as appropriate for inter-zone or intra-zone traffic including TCP stateful inspection, syn-attack guard, policy-based control, load balancing and other functionalities on each data stream. In one implementation, the packet switched communication system can be configured to include multiple IEEE 802.1Q VLAN based L2 transparent domains. A user can create multiple VLANs, each having its own policy for firewall control. In addition, methods are provided for VPN tunnel capability to connect remote clients to the L2 domain. Methods are provided to guard against broadcasting information throughout the zones and violating one or more security constraints when a MAC address that is being processed is not recognized. The methods include the broadcast of probe packets to discover topology information for unknown MAC destinations.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. **1** is a block diagram of a packet switched communication system including an L2 firewall enabled switch.

FIG. **2***a* is a schematic view of an L2 firewall enabled switch.

FIG. **2***b* shows an exemplary communication network including plural zones partitioned by a single security switch.

FIG. **3** is a flow diagram of a method for processing packets in the security switch of FIG. **2***a*.

FIG. **4** is a flow diagram for a method for processing un-recognized packets in the security switch of FIG. **2***a*.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Referring now to FIG. **1**, a packet switch communication network **100** includes a plurality of terminal units **102** configured in a plurality of zones **104** and coupled by one or more switches **106**.

In one implementation, each terminal unit **102** is of the form of a standalone computer (e.g., a personal computer, a laptop or workstation). Alternatively, one or more terminal units may be of the form of a personal digital assistant (PDA), Web pad, two-way pager, cellular handset, or other termination or remote device in a communication or computing environment. In one implementation, each terminal is a gateway to another network or group of terminal units (e.g., to a LAN or a pool of servers).

Each zone **104** embodies a security domain in the communication system. Each security domain can include separate policy, traffic management, accounting and administrative definitions and functions. Security policies, traffic management and other filtering functions can be enforced among and within zones. In one implementation, security policies are enforced between zones, while intra-zone communications are not subject to the security constraints. In one implementation, zones overlap. When zones overlap, policies associated with a parent zone can be a superset of the policies associated with one or more sub-zones (each including a subset of the overall policies). Alternatively, the policies associated with the parent zone may be separate and distinct from the policies of each sub-zone. For example, in one implementation, a zone can include one or more sub-zones, each including a separate set of policies.

In one implementation, each zone is associated with physical boundaries or other segmentation in the communication network. Alternatively, the assignment of particular terminal units to zones may represent groupings or combinations in a business structure (e.g., zones used to separate different functional entities in a business organization). Alternatively, the zones have no particular relation to physical boundaries. Communication between terminal units in each zone and among terminal units within a zone are controlled in accordance with protocols described below in association with switch **106**.

Switch **106** may be of different types. In one implementation, each switch **106** is configured as a layer 2 (L2) device and includes a plurality of ports on which packets from the communication network are received and transferred in accordance with L2 protocols. Each switch **106** includes a media access connection (MAC) table for use in determining

switching of received packets. The MAC table associates MAC addresses with ports of the switch **106**. Packets are processed as they arrive at the ports of each switch **106** in accordance with L2 header information contained within a given packet. Depending on the MAC address, packets are switched to an appropriate output port as specified in the MAC table.

One or more of switches **106** are configured to enforce security domain constraints. For example, one or more of switches **106** is configured as an L2 firewall enabled security switch (hereinafter "security switch"). Referring now to FIG. **2**, a security switch **200** includes a plurality of ports **202**, a switch fabric **220** and an L2 controller **230**. Each port **202** is coupled to a security controller **204** by a bus **206**. The security controller **204** is coupled to one or more storage elements **208**. In one implementation (not shown), each port **202** is associated with a separate security controller **204** and storage element **208**. Alternatively, the security controller functionality can be combined in a single (as shown) or lesser number of individual security controller units. In addition, packets associated with all ports **202** can be stored in a single memory element **208** (as shown). Security switch **200** also includes a firewall device **210** that is coupled to (each) storage element **208** by a security bus **211**.

L2 controller **230** supports L2 switching protocols. Packets are either directly processed (e.g., intra-zone packets) or processed after a security screening (e.g., for inter-zone packets) as discussed in greater detail below. Associated with L2 controller **230** is a MAC table **235**. MAC table **235** includes plural entries each of which includes a MAC address and an indicator of a port **202** associated therewith. Switch fabric **220** is used to route traffic from storage element **208** to a respective port **202** under the control of L2 controller **230** using bus **221**.

Storage element **208** is partitioned into two portions. A first portion **215** is used to store packets received from a port **202** that are not subject to security screening. For example, in one implementation, packets received from a terminal unit in a same security zone (e.g., intra-zone traffic) are not subject to security screening. Un-screened packets are processed directly by L2 controller **230** and forwarded out a designated port in accordance with L2 protocols as specified in MAC table **235**. Second portion **217** is used to store packets to be screened by firewall device **210**.

Security controller **204** includes a screening engine **240**. Screening engine **240** examines each packet received from a respective port **202** and determines whether security screening is to be performed. In one implementation, screening engine **240** examines the L2 header for each packet, and based on the screening, either forwards the packet to the first or second portion **215** and **217**, respectively, of storage element **208**. The L2 header includes a destination MAC address that can be mapped to an egress port on the device using the MAC table **235**. Associated with each ingress and egress port is a security zone identifier. Security zone identifiers can be stored in a table of zone identifiers (not shown) that is indexed by port identifier (id). Screening engine **240** compares the security zone identifier associated with the packet being processed (determined from the identification of the egress port from the MAC table using the destination MAC address in the header of the packet being processed) with the security zone identifier associated with the port on which the packet was received in the device. Based on the comparison, screening engine **240** can determine whether the packet is destined for another zone (i.e., constitutes intra-zone or inter-zone communication).

The screening of packets can be with or without the knowledge of the individual terminal units. Associated with security switch **200** is a user interface (not shown) and associated management tools (not shown) for constructing one or more security zones. In one implementation, the security zones are determined based on the destination MAC address included in the L2 header of the packet received. More specifically, each egress port can be assigned to a security zone and have an associated security zone identifier associated therewith. Alternatively, the security zones can be created for plural users coupled to different ports of the security switch **200**. For example, security switch **200** can be configured to include three ports, where terminal units associated with a first two of the ports are assigned to a first zone, while terminal units associated with the third port are assigned to a second zone. Other configurations are possible. Zone assignments and partitions are discussed in greater detail below. The user interface allows an administrator or user to configure the security switch **200**. The security switch **200** can be configured to create plural security zones and associate one or more interfaces with each zone. Thereafter, policies can be established for inspecting or otherwise screening packets as they traverse the security switch **200**.

Firewall device **208** includes plural engines for performing packet screening prior to routing packets through security switch **200**. Firewall device **208** includes a firewall engine **270** and associated policies **271**, authentication engine **272**, encryption engine **274**, decryption engine **276** and a firewall controller **278**.

Firewall controller **278** extracts packets from second portion **217** of storage element **208**. Firewall controller **278** oversees the distribution of packets within the firewall device as well as the coordination among the respective engines. Each packet is evaluated and processed in accordance with policies based on one or more considerations. For example, packets can be screened based on source, destination or both. One or more policies **271** are retrieved and used by firewall engine **270** to inspect the packet. Packet inspection may also require encryption, decryption and authentication services. One or more of the encryption **272**, decryption **274** and authentication **276** engines can be invoked by the firewall controller **278** as part of the inspection processes. In addition, other services can be provided including virtual private network termination services, session set-up and various other traffic management and security related functions. Examples of screening services are discussed in greater detail below. After the inspection, packets can be forwarded in the network or dropped as appropriate. In one implementation, packets that are to be forwarded (e.g., pass the inspection) are prepared as appropriate (e.g., encrypted) then forwarded to the first portion **215** of storage element **208**. Alternatively, the packets may be returned to the second portion **217** of storage element **208** and marked as having been screened. In one implementation, screened packets are forwarded to a queue for processing by L2 controller **230**. Screened packets are then processed by L2 controller **230** and switched to an appropriate output port in accordance with conventional L2 processing protocols.

Each security switch **200** can be configured to create plural security zones. For example, a communications network having a security switch **200** is shown in FIG. **2***b*. The communications network is a VLAN structure that includes 3 zones. Security switch **200** includes a user interface and administrative control mechanisms for creating each of the security zones, specifying policies and other criteria for defining and managing each zone. The security zones enforced by the security switch **200** can be transparent to the end users. That is, the security zones can be established at the security switch **200** including the specification of all operating parameters associated with the security domain. Users in each zone may be unaware of the zone structure and may communicate with other users in a conventional manner. For example, a virtual private network can be created between users including encryption and decryption services without requiring the actual encryption and decryption support in the respective end users (e.g., encryption and decryption services can be provided in secure switches disposed between the two users). Accordingly, a system administrator can create a virtual private network between a remote user in one security zone and another user in a second security zone where the individual users are unaware of the VPN services and are not required to include encryption or decryption services locally. In one implementation, the administrator provisioned VPN services are specified for remote users in a same zone.

Alternatively, the users may be aware of the security structure and include indicators (e.g., zone identifiers) in packets transferred to other users. Each user may define their own custom L2 zone and an inter-zone policy for their network security requirements. For example, security switch **200** shown in FIG. **2***b* embodies a VLAN that includes v1-trust, v1-untrust and v1-dmz zones. V1-trust defines a zone that includes two users including user **291** and user **292**. V1-untrust defines a zone that includes a single user **293**. V1-dmz defines a zone that includes three users, users **291**, **292** and user **294**. Separate policies can be enforced for communications between the three zones. For example, communications that are intra-zone between user **291** and user **292** will not require inspection, and as such are handled by security switch **200** in accordance with conventional L2 protocols.

Communications from user **291** to user **293** will invoke an inspection process as defined by the security system architect (e.g., user **291** or **292** or an administrator for such) for communications between V1-trust and V1-untrust. Similarly, communications between user **294** and user **291** will invoke an inspection process (e.g., a potentially lesser screen) for communications between V1-dmz and V1-trust.

Multiple interfaces are allowed inside each zone. For intra-zone traffic, security switch **200** behaves like a tradition L2 bridge forwarding a given packet based on the destination MAC-address. In one implementation, no firewall protection mechanisms are applied for the intra-zone traffic.

For inter-zone traffic, standard firewall inspections (including policy inspection, TCP stateful inspection, etc. as described above) are performed for each incoming packet. In all cases, the egress interface is determined by the learned destination MAC address on the interface.

Packet Flow

Referring now to FIG. **3**, a method **300** is shown, as invoked by the security switch **200**, for processing packets. The method described is made with no particular reference to the specific hardware elements performing the steps. An exemplary hardware configuration is given above. The method can however be implemented in L2 switches having other configurations. The method begins with the receipt of a packet (**302**). The packet is evaluated to determine whether the packet is to be inspected (**304**). If so, the packet is pre-processed as appropriate (**305**) and one or more policies are retrieved (**306**). The pre-processing of the packet can include decryption and authentication services. The retrieval of a policy includes the identification of the zone to which the packet is being transferred. Packets traveling between zones can be inspected using a security policy. Intra-zone communications may not be inspected. In one implementation, policies can be enforced on intra-zone communications. The retrieval of a policy includes a MAC look-up for the MAC destination address in a received packet in the MAC table to determine an egress port associated with the MAC address and necessarily a security zone. The security zones associated with the packet's ingress and egress ports are compared to determine if the packet is passing to another zone. Assuming that an inspection is to occur, an appropriate policy is retrieved (i.e., based on the ingress port and egress port identifiers and their respective security zones). Thereafter, the packet is inspected (**308**). Packet inspection can include screening and dropping the packet as required. If the packet is to be forwarded on the network (**309**), post-processing operations are invoked as appropriate (**310**). Alternatively, the packet is dropped (**311**). The post processing operations can include session set-up, encryption and other functions. Thereafter the packet is processed in accordance with conventional L2 protocols starting at step **312**.

At step **312**, either a packet has passed inspection or did not require inspection. In either case, L2 header information is extracted to determine a MAC address associated with the packet. A look-up of the MAC address is performed (**314**) and the packet is then routed to an appropriate output port (**316**). Thereafter the process ends.

Referring again to FIG. **2**, the process steps are described with reference to one hardware implementation of the invention. Packets are received at a port **202**. Each packet is transferred on bus **205** to, and routed through, security controller **204** and stored in storage element **208** via a storage bus **209**. Security controller **204** evaluates each packet to determine if inspection is required and forwards the packets to an appropriate portion of storage device **208**. Packets that are not to be inspected (i.e., packets stored in first portion **215** of storage device **208**) are processed by L2 controller **230**. When L2 controller **230** is available, packets are fetched and processed to determine a port to which the packet should be forwarded. L2 controller **230** evaluates the MAC address associated with the packet, and using MAC table **235**, determines a port for routing. After processing by the L2 controller **230**, the packet is forwarded to an appropriate link into switch fabric **220** for routing to a determined output port **202**.

Packets that are to be inspected are transferred by security controller **204** into second portion **217** of storage element **208**. When firewall engine **230** is available, a packet is fetched and processed to determine a security policy to be used in inspecting the packet. Firewall engine **270** evaluates IP address(es) associated with the packet and implements traffic control and management functions as appropriate. Packets that are to be forwarded (i.e., pass inspection) are returned to storage element **208**. Thereafter, the packet can be forwarded to an appropriate link into switch fabric **220** for routing to a determined output port **202**. Other packets are dropped or otherwise handled in accordance with the policies defined for the given security zones.

As discussed above, all packets that pass the inspection in the firewall device **210** as well as all packets that are not required to be inspected, are processed by L2 controller **230** in accordance with conventional L2 protocols. In one implementation, the processing of packets by L2 controller is modified to maintain security zones. More specifically, as discussed above, conventional L2 switches broadcast on all ports a packet that has a MAC address that is not recognized. This type of broadcast may well violate one or more security policies in place for given zones in the communication network. Accordingly, in one implementation a test packet is broadcast to each port. The broadcasting of test packets is described in more detail in association with FIG. **4**.

Referring now to FIG. **4**, a method **400** is shown for handling packets by the L2 controller and includes receiving a packet to be processed (**402**). The MAC address for the packet is extracted (**404**). A check is made to locate an entry in a MAC address table that corresponds to the extracted MAC address (**406**). If a match is located (**407**), the packet is routed to an output port associated with the matching entry (**408**). If no match is located, the packet is dropped (**410**). In one implementation, the packet is merely held for a predetermined amount of time in hope of receiving information regarding the non-matching MAC address. If no match is located, a probe packet is created (**412**). The probe packet includes the MAC address associated with the packet being processed (i.e., the original ingress packet). In one implementation, the probe packet is an "ICMP PING" packet with an IP TTL field set to 1. Each packet includes the same MAC addresses (L2) and source/destination IPs (L3) as the ingress packet whose MAC address could not be located. The probe packet is then broadcast to all ports (**414**). A check is made to determine if a response is received on any of the security device's ports (**416**). The ICMP PING packet will cause the right gateway, which was to receive and forward the original ingress packet, to respond to the L2 controller in the device with an "ICMP TTL expired" message packet. From the expired packet, the system can identify the proper egress port/zone associated with the received MAC address. This method guarantees that no information in the original ingress packet will be leaked out. If a response is received (indicating that a device coupled to the receiving port is configured to process packets having the identified MAC address), then the MAC table is updated to include an entry having the MAC address and a port identifier indicating the port on which the response was received (**418**). Thereafter the process ends.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the firewall device has been described in terms of screening at the L3 layer level. Alternatively, other screening can be invoked at other levels including layers up to and including layer 7 (L7) processing. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. In a network device, a method comprising:

receiving a packet via a network that includes a plurality of distinct security domains;

determining whether the packet is to remain within a first one of the distinct security domains or pass between two of the distinct security domains;

performing, based on a first determination that the packet is to pass between the two distinct security domains security, security screening on the packet before routing the screened packet to an egress port of the network device for forwarding on the network; and

routing, based on a second determination that the packet is to remain within the first distinct security domain, the packet to an egress port of the network device for forwarding on the network without performing the security screening on the packet.

2. The method of claim **1**, where performing the security screening comprises:

enforcing at least one security constraint with respect to the screened packet.

3. The method of claim **2**, where the enforcing the at least one security constraint comprises at least one of applying a security policy, performing traffic management, or performing a filtering function.

4. The method of claim **2**, where the enforcing the at least one security constraint comprises at least two of applying a security policy, performing traffic management, or performing a filtering function.

5. The method of claim **2**, where the enforcing the at least one security constraint comprises applying a security policy, performing traffic management, and performing a filtering function.

6. The method of claim **1**, further comprising:

creating, at the network device, the plurality of distinct security domains.

7. The method of claim **6**, where the creating comprises specifying policies and other criteria for defining and managing each of the plurality of distinct security domains.

8. The method of claim **6**, where operating parameters associated with each of the plurality of distinct security domains are not made known to users associated with the plurality of distinct security domains.

9. The method of claim **1**, where operating parameters associated with each of the plurality of distinct security domains are defined by users associated with each of the plurality of distinct security domains.

10. The method of claim **1**, where the determining whether the packet is to remain within the first distinct security domain or pass between the two distinct security domains comprises:

comparing a security domain identifier associated with an ingress port on which the packet was received to a security domain associated with an egress port on which the packet is to be forwarded.

11. The method of claim **1**, where the determining whether the packet is to remain within the first distinct security domain or pass between the two distinct security domains comprises:

identifying a security domain indicator in the packet.

12. A network device comprising:

an ingress port to receive a packet via a network that includes a plurality of distinct security domains;

a controller to determine whether the network device is to transfer the packet within a first one of the distinct security domains or between two of the distinct security domains;

a security device to perform security screening, based on a first determination that the packet is to be forwarded between the two distinct security domains security, on the packet before routing the packet to an egress port of the network device for forwarding on the network; and

an engine to route the packet, based on a second determination that the packet is to be forwarded within the first distinct security domain, to an egress port of the network device for forwarding on the network without performing the security screening on the packet.

13. The network device of claim **12**, where the security device is configured to enforce at least one security constraint with respect to the packet.

14. The network device of claim **13**, where the enforcing the at least one security constraint comprises at least one of applying a security policy, performing traffic management, or performing a filtering function.

15. The network device of claim **13**, where the enforcing the at least one security constraint comprises at least two of applying a security policy, performing traffic management, or a filtering function.

16. The network device of claim **13**, where the enforcing the at least one security constraint comprises applying a security policy, performing traffic management, and performing a filtering function.

17. The network device of claim **12**, further comprising:

a user interface to be used to create the plurality of distinct security domains.

18. The network device of claim **17**, where creating the plurality of distinct security domains comprises specifying policies and other criteria for defining and managing each of the plurality of distinct security domains.

19. The network device of claim **17**, where operating parameters associated with each of the plurality of distinct security domains are not made known to users associated with the plurality of distinct security domains.

20. The network device of claim **12**, where operating parameters associated with each of the plurality of distinct security domains are defined by users associated with each of the plurality of distinct security domains.

21. The network device of claim **12**, where the controller is configured to determine whether the network device is to transfer the packet within the first distinct security domain or between the two distinct security domains comprises by comparing a security domain identifier associated with an ingress port on which the packet was received to a security domain associated with an egress port on which the packet is to be forwarded.

22. The network device of claim **12**, where the controller is configured to determine whether the network device is to transfer the packet within the first distinct security domain or between the two distinct security domains comprises by identifying a security domain indicator in the packet.

23. A system comprising:

one or more devices comprising:

means for receiving a packet via a network that includes a plurality of distinct security domains;

means for determining whether the packet is to be forwarded over the network within a first one of the distinct security domains or between two of the distinct security domains;

means for performing security screening on the packet based on a first determination that the packet is to be forwarded between the two distinct security domains security;

means for forwarding the screened packet on the network or dropping the packet based on the security screening; and

means for forwarding the packet on the network without performing the security screening on the packet based on a second determination that the packet is to be forwarded within the first distinct security domain.

* * * * *

EXHIBIT C

US007650634B2

(12) **United States Patent**
Zuk

(10) **Patent No.: US 7,650,634 B2**
(45) **Date of Patent: Jan. 19, 2010**

(54) **INTELLIGENT INTEGRATED NETWORK SECURITY DEVICE**

(75) Inventor: **Nir Zuk**, Redwood City, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 727 days.

(21) Appl. No.: **10/402,920**

(22) Filed: **Mar. 28, 2003**

(65) **Prior Publication Data**

US 2004/0030927 A1 Feb. 12, 2004

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/072,683, filed on Feb. 8, 2002.

(51) **Int. Cl.**

G06F 9/00	(2006.01)
H04L 29/06	(2006.01)
G06F 15/173	(2006.01)

(52) **U.S. Cl.** **726/13**; 713/152; 713/161; 709/226

(58) **Field of Classification Search** 713/189, 713/152, 161; 726/13; 709/226
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,598,410	A *	1/1997	Stone	370/469
5,606,668	A	2/1997	Shwed	395/200.11
5,835,726	A	11/1998	Shwed et al.	395/200.59
6,006,264	A *	12/1999	Colby et al.	709/226
6,052,788	A	4/2000	Wesinger, Jr. et al.	
6,119,236	A	9/2000	Shipley	713/207
6,205,551	B1	3/2001	Grosse	
6,253,321	B1	6/2001	Nikander et al.	713/160
6,275,942	B1	8/2001	Bernhard et al.	713/201
6,279,113	B1	8/2001	Vaidya	713/201
6,301,668	B1	10/2001	Gleichauf et al.	713/201
6,304,975	B1	10/2001	Shipley	713/201
6,311,278	B1	10/2001	Raanan et al.	713/201
6,321,338	B1	11/2001	Porras et al.	713/201
6,370,603	B1	4/2002	Silverman et al.	
6,421,730	B1	7/2002	Narad et al.	709/236
6,449,647	B1 *	9/2002	Colby et al.	709/226
6,453,345	B2	9/2002	Trcka et al.	709/224

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1 143 660 A2	10/2001

(Continued)

OTHER PUBLICATIONS

International Search Report for corresponding PCT application, PCT/US2004/009607, dated Oct. 22, 2004, 3 pages.

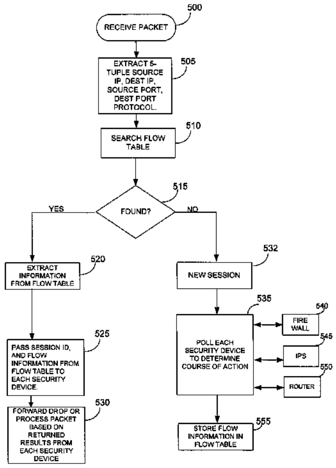
(Continued)

Primary Examiner—Nasser G Moazzami
Assistant Examiner—Mohammad W Reza
(74) *Attorney, Agent, or Firm*—Harrity & Harrity, LLP

(57) **ABSTRACT**

Methods, computer program products and apparatus for processing data packets are described. Methods include receiving the data packet, examining the data packet, determining a single flow record associated with the packet and extracting flow instructions for two or more devices from the single flow record.

41 Claims, 9 Drawing Sheets


500
RECEIVE PACKET
505
EXTRACT 5-TUPLE SOURCE IP, DEST IP, SOURCE PORT, DEST PORT PROTOCOL.
510
SEARCH FLOW TABLE
515
FOUND?
YES
NO
520
EXTRACT INFORMATION FROM FLOW TABLE
525
PASS SESSION ID, AND FLOW INFORMATION FROM FLOW TABLE TO EACH SECURITY DEVICE.
530
FORWARD DROP OR PROCESS PACKET BASED ON RETURNED RESULTS FROM EACH SECURITY DEVICE
532
NEW SESSION
535
POLL EACH SECURITY DEVICE TO DETERMINE COURSE OF ACTION
540
FIRE WALL
545
IPS
550
ROUTER
555
STORE FLOW INFORMATION IN FLOW TABLE

U.S. PATENT DOCUMENTS

6,466,985	B1	10/2002	Goyal et al.	709/238
6,487,666	B1	11/2002	Shanklin et al.	713/201
6,499,107	B1	12/2002	Gleichauf et al.	713/201
6,600,744	B1	7/2003	Carr et al.	
6,704,278	B1	3/2004	Albert et al.	
6,768,738	B1	7/2004	Yazaki et al.	
6,781,992	B1	8/2004	Rana et al.	
6,788,648	B1 *	9/2004	Peterson	370/252
6,851,061	B1	2/2005	Holland et al.	726/23
6,856,991	B1 *	2/2005	Srivastava	707/10
6,981,158	B1	12/2005	Sanchez et al.	
7,006,443	B2 *	2/2006	Storr	370/236.1
7,376,085	B2	5/2008	Yazaki et al.	
2001/0028650	A1	10/2001	Yoshizawa et al.	
2002/0032797	A1 *	3/2002	Xu	709/238
2002/0080789	A1	6/2002	Henderson et al.	
2002/0124187	A1	9/2002	Lyle et al.	713/201
2002/0126621	A1	9/2002	Johnson et al.	
2002/0161839	A1	10/2002	Colasurdo et al.	
2003/0105976	A1	6/2003	Copeland	713/201
2003/0145225	A1	7/2003	Bruton et al.	713/201
2003/0149887	A1	8/2003	Yadav	
2003/0149888	A1	8/2003	Yadav	713/200
2005/0141503	A1	6/2005	Welfeld	
2005/0163132	A1	7/2005	Mieno et al.	

FOREIGN PATENT DOCUMENTS

EP	1 427 162	A1	6/2004
JP	10-107795		4/1998
JP	11-316677		11/1999
JP	2000-312225		11/2000
JP	2001-313640		11/2001
JP	2003-78549		3/2003
WO	WO 03/025766	A1	3/2003
WO	WO 03/061238	A2	7/2003

OTHER PUBLICATIONS

Co-pending U.S. Appl. No. 10/961,075, filed Oct. 12, 2004, entitled "Intelligent Integrated Network Security Device for High-Availability Applications," Nir Zuk et al., 24 page specification, 13 sheets of drawings.

Co-pending U.S. Appl. No. 10/072,683, filed Feb. 8, 2002, entitled "Multi-Method Gateway-Based Network Security Systems and Methods," Nir Zuk et al., 62 page specification, 16 sheets of drawings.

Stonesoft, 'StoneBeat *Security Cluster White Paper*,' Aug. 2000, Finland, pp. 1-9.

Stonesoft, '*Secure Highly Available Enterprise-A White Paper*,' Feb. 2001, Finland, pp. 1-10.

Stonesoft, '*StoneGate White Paper*,' Mar. 2001, Finland, pp. 1-6.

Stonesoft Corp. '*StoneGate*,' product webpage, www.stonesoft.com/document/363.html, Mar. 27, 2001 (print date), pp. 1-2.

Stonesoft Corp. '*Next Level of Network Accessibility*' webpage, www.stonesoft.com/document/183.html, Mar. 27, 2001 (print date), p. 1.

Stonesoft Corp., '*Platforms*,' webpage, www.stonesoft.com/document.186/html, Mar. 27, 2001 (print date), p. 1.

Nokia, 'Technical White Paper: The IP Clustering Power of Nokia VPN-Keeping Customers Connected,' Apr. 2001, pp. 1-13.

Nokia, '*Nokia VPN Solutions—Nokia VPN CC2500 Gateway*,' 2001, product information, pp. 1-2.

Nokia, '*Nokia VPN Solutions—Nokia VPN CC5200 Gateway*,' 2001, product information, pp. 1-2.

Nokia, '*Nokia VPN Solutions—Nokia VPN CC5205 Gateway*,' 2001, product information, pp. 1-2.

Axelsson, S., "Intrusion Detection Systems: A Survey and Taxonomy," *Dept. of Computer Eng*., Chalmers Univ. of Technology, Goteborg, Sweden, Mar. 14, 2000, pp. 1-27.

Avolio, F., "Firewalls and Virtual Private Networks," CSI Firewall Archives, printed Nov. 13, 2001, URL: http://www.spirit.com/CSI/Papers/fw+vpns.html, pp. 1-7.

Bace, R., "An Introduction to Intrusion Detection & Assessment," ICSA Intrusion Detection Systems Consortium White Paper, 1999, URL: http://www.iscalabs.com/html/communities/ids/whitepaper/Intrusion1.pdf, pp. 1-38.

Business Wire, Inc., "NetScreen and OneSecure Unite to Deliver Industry's First Total Managed Security Services Platform," San Jose, CA, Feb. 20, 2001, pp. 1-2.

Business Wire, Inc., "OneSecure Launches the First Co-Managed Security Services Platform," Denver, CO, Jan. 29, 2001, pp. 1-2.

Carr, Jim, "Intrusion Detection Systems: Back to Front?," *Network Magazine*, Sep. 5, 2001, URL: http://www.networkmagazine.com/article/NMG20010823S0007/2, pp. 1-9.

Check Point Software Technologies Ltd., Firewall-1® Technical Overview P/N 500326, www.checkpoint.com, Oct. 2000, pp. 1-29.

Cisco Systems, "Cisco IOS Firewall Intrusion Detection System," Cisco IOS Release 12.0(5)T, 2001, pp. 1-40.

Cisco Systems, "Cisco IOS Firewall Authentication Proxy," Cisco IOS Release 12.0(5)T, 2001, pp. 1-48.

Clark, D., "RFC815-IP Datagram Reassembly Algorithms," Internet RFC/STD/FYI/BCP Archives, http://www.faqs.org/rfcs/rfc815.html, Jul. 1982, pp. 1-8.

Copeland, Dr. John A., "Observing Network Traffic-Techniques to Sort Out the Good, the Bad, and the Ugly," PowerPoint Slide Presentation presented to ISSA-Atlanta, Jun. 27, 2001, pp. 1-22.

Denning, Dorothy E., "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, No. 2, Feb. 1987, 17 pages.

Farrow, Rik, "An Analysis of Current Firewall Technologies," *CSI 1997 Firewalls Matrix*, 1998, URL: http://www.spirit.com/CSI/Papers/farrowpa.htm, pp. 1-5.

Firewall Product Comparison Table: VelociRaptor, BorderWare Firewall Server and Firewall-1/VPN-1 Gateway, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: PIX Firewall, CyberGuard Firewall for UnixWare & CyberGuard Firewall for Windows NT, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: CyberGuard Premium Appliance Firewall, InstaGate EX & BizGuardian VPN Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Server Protector 100, GNAT Box Firewall Software & Lucent Managed Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: Internet Security and Acceleration (ISA) Server 2000, NetBSD/i386 Firewall & Guardian Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: NetScreen-10 and NetScreen-100, CyberwallPLUS & BorderManager, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: Gauntlet Firewall, Barricade Classic/XL & Barricade S, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Sidewinder™, SecurePipe Managed Firewall Service & SnapGear, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: SonicWALL PRO, Sunscreen Secure Net & WinRoute Pro 4.1, www.spirit.com, printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: WatchGuard Technologies, Inc. LiveSecurity System 4.6, www.spirit.com, printed Nov. 13, 2001, pp. 1-4.

Graham, R., "FAQ: Network Intrusion Detection System," www.robertgraham.com/pubs/network-intrusion-detection.html, Ver. 0.8.3, Mar. 21, 2000, pp. 1-43.

Habra, N. et al., "ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis," Proceedings of the ESORICS '92, European Symposium on Research in Computer Security, Nov. 23-25, 1992, Toulouse, Springer-Verlag, 16 pages.

ICSA Labs, Intrusion Detection System Buyer's Guide, ICSA White Paper, 1999, pp. 1-52.

Jackson, K. et al., "Intrusion Detection System (IDS) Product Survey," *Los Alamos National Laboratory*, Los Alamos, NM, LA-UR-99-3883 Ver. 2.1, Jun. 25, 1999, pp. 1-103.

Jones, Kyle, "Introduction to Firewalls," *IT Audit.org Forum Network Management*, vol. 2, May 1, 1999, URL: http://www.itaudit.org/forum/networkmanagement/f209nm.htm, pp. 1-5.

Lancope, "The Security Benefits of a Flow-Based Intrusion Detection System," White Paper, date unknown, pp. 1-11.

LapLink, Inc., "Article #178-Introduction to Firewalls," www.laplink.com/support/kb/article.asp?ID=178, Apr. 24, 2001, pp. 1-3.

McHugh, J. et al., "Defending Yourself: The Role of Intrusion Detection Systems," *Software Engineering Institute*, IEEE Software Eng., Sep./Oct. 2000, pp. 42-51.

Network ICE Corporation, "Why Firewalls Are Not Enough," at www.networkice.com/products/firewalls.html, 2000, pp. 1-9.

Power, R., et al., "CSI Intrusion Detection System Resource-Five Vendors Answer Some No-Nonsense Questions on IDS," *Computer Security Alert #184*, Jul. 1998, pp. 1-8.

Power, R., "CSI Roundtable: Experts discuss present and future intrusion detection systems," *Computer Security Journal*, vol. XIV, #1, URL: http://www.gocsi.com/roundtable.htm, 2001, pp. 1-20.

Sample, Char, et al., "Firewall and IDS Shortcomings," SANS Network Security, Monterey, CA, Oct. 2000, pp. 1-13.

Smith, Gary, "A Brief Taxonomy of Firewalls-Great Walls of Fire," SANS Institute's Information Security Reading Room, May 18, 2001, URL: http://www.sans.org/infosecFAQ/firewall/taxonomy.htm, pp. 1-21.

Spitzner, Lance, "How Stateful is Stateful Inspection? Understanding the FW-1 State Table," http://www.enteract.com/~1spitz/fwtable.html, Nov. 29, 2000, pp. 1-8.

Sundaram, A., "An Introduction to Intrusion Detection," www.acm.org/crossroads/xrds2-4/intrus.html, Jan. 23, 2001, pp. 1-12.

Tyson, Jeff, "How Firewalls Work," http://www.howstuffworks.com/firewall.htm/printable, 2001, pp. 1-7.

Xinetica, Ltd., "An Overview of Intrusion Detection Systems," Xinetica White Paper, Nov. 12, 2001 (print date), URL: http://www.xinetica.com/tech_explained/general/ids/wp_ids.html, pp. 1-9.

Zuk, Nir, "Protect Yourself With Firewalls," www.techtv.com, Jul. 12, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3325761,00.html, pp. 1-3.

Zuk, Nir, "How the Code Red Worm Works," www.techtv.com, Sep. 21, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3349133,00.html, pp. 1-2.

Petersen, S., et al., "Web apps pose security threat," ZDNet: Tech Update, Jan. 29, 2001, URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2679177,00.html, pp. 1-3.

Lancope, "StealthWatch Provides Early Detection of the Code Red Worm and its Future Variants," www.stealthwatch.com, date unknown, pp. 1-4.

Reavis, J., "Cash and Burn," Jun. 2001, 6 pages.

SOS Corporation, "An Introduction to Firewalls," 1995, URL: http://www.uclan.ac.uk/facs/destech/compute/staff/haroun/FIREWALS.htm, pp. 1-3.

Morgan, Lisa, "Be Afraid, Be Very Afraid," InternetWeek Intrusion Detection Systems, Jan. 3, 2001, pp. 1-6.

Mullins, Robert, "'Cyber war' raises security concerns," *Silicon Valley/San Jose Business Journal*, May 11, 2001, pp. 1-4.

James P. Anderson Co., "Computer Security Threat Monitoring and Surveillance," Apr. 15, 1980, 56 pages.

Internet Security Systems, Inc., "Realsecure™, The RealSecure Advantage," 2001, 2 pages.

Chuvakin, A., et al., "Basic Security Checklist for Home and Office Users," SecurityFocus, Nov. 5, 2001, pp. 1-5.

Network Ice, "SMTP WIZ command," 2001, URL: http://networkice.com/Advice/Intrusions/2001006/default.htm, pp. 1-2.

Bace, R., et al., "NIST Special Publication on Intrusion Detection Systems," National Institute of Standards and Technology Special Publication, date unknown, pp. 1-51.

G. Navarro: A Partial Deterministic Automaton for Approximate String Matching, 1997, Department of Computer Science, University of Chile, 13 pages.

G. Navarro et al.: Improving an Algorithm for Approximate Pattern Matching, 1998, Department of Computer Science, University of Chile, 34 pages.

Network Magazine, vol. 2, No. 2, pp. 116-119 (with English abstract).

Software Design, Nov. 1996, pp. 39-58 (with English abstract).

* cited by examiner

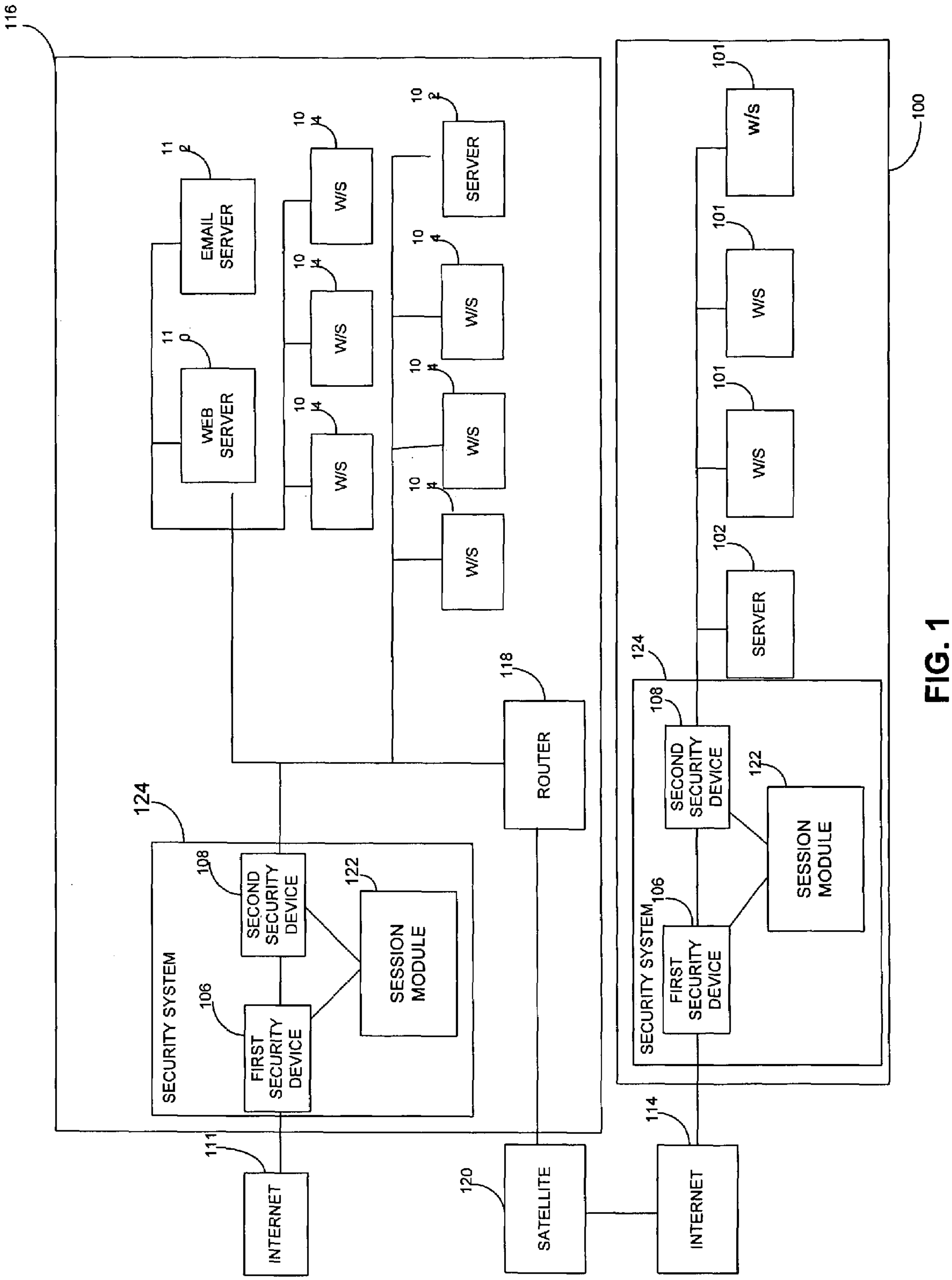

116
SECURITY SYSTEM
124
FIRST SECURITY DEVICE
106
SECOND SECURITY DEVICE
108
SESSION MODULE
122
WEB SERVER
11 0
EMAIL SERVER
11 2
W/S
10 4
W/S
10 4
W/S
10 4
W/S
10 4
W/S
10 4
W/S
10 4
SERVER
10 2
ROUTER
118
INTERNET
111
SATELLITE
120
INTERNET
114
100
SECURITY SYSTEM
124
FIRST SECURITY DEVICE
106
SECOND SECURITY DEVICE
108
SESSION MODULE
122
SERVER
102
W/S
101
W/S
101
W/S
101

FIG. 1

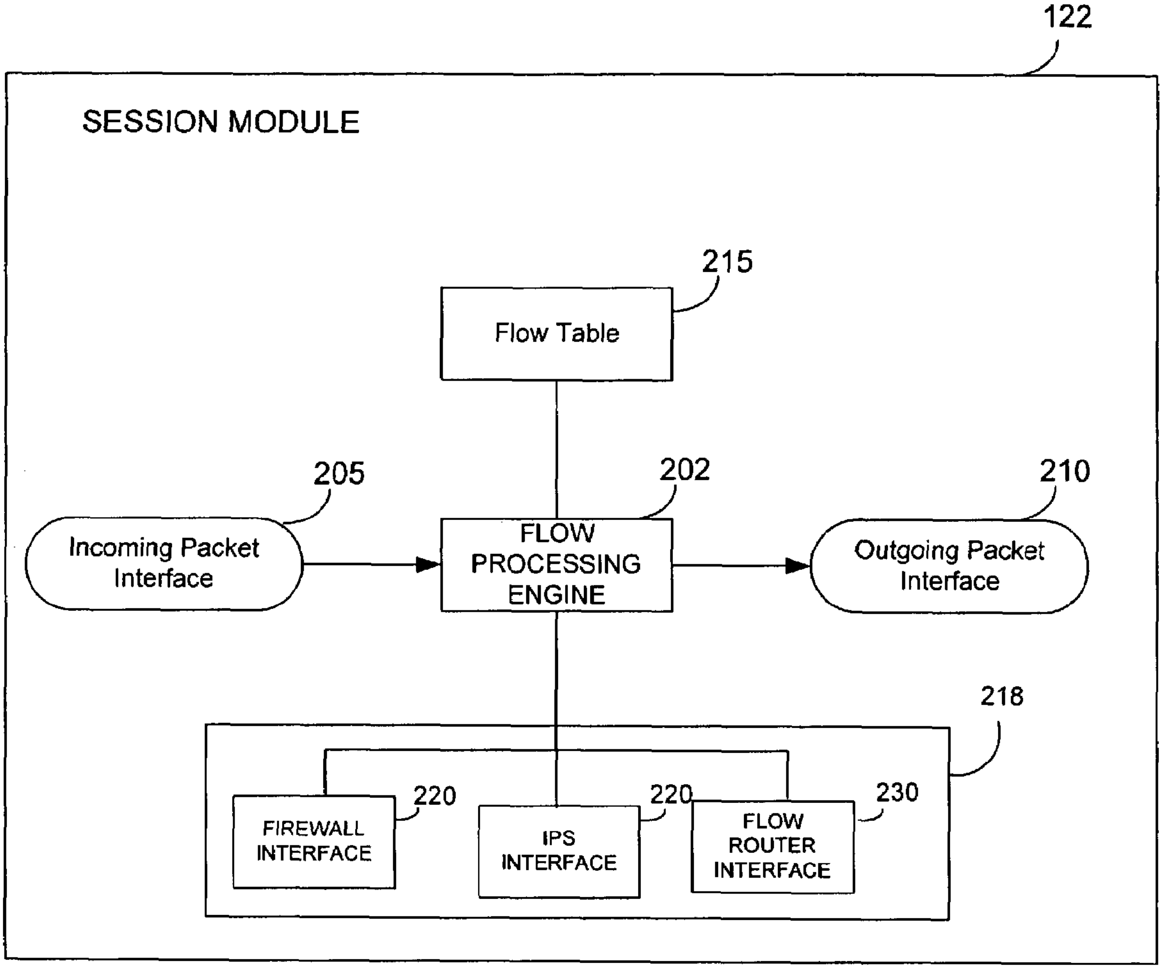

122
SESSION MODULE
215
Flow Table
205
Incoming Packet Interface
202
FLOW PROCESSING ENGINE
210
Outgoing Packet Interface
218
FIREWALL INTERFACE
220
IPS INTERFACE
220
FLOW ROUTER INTERFACE
230

FIG. 2

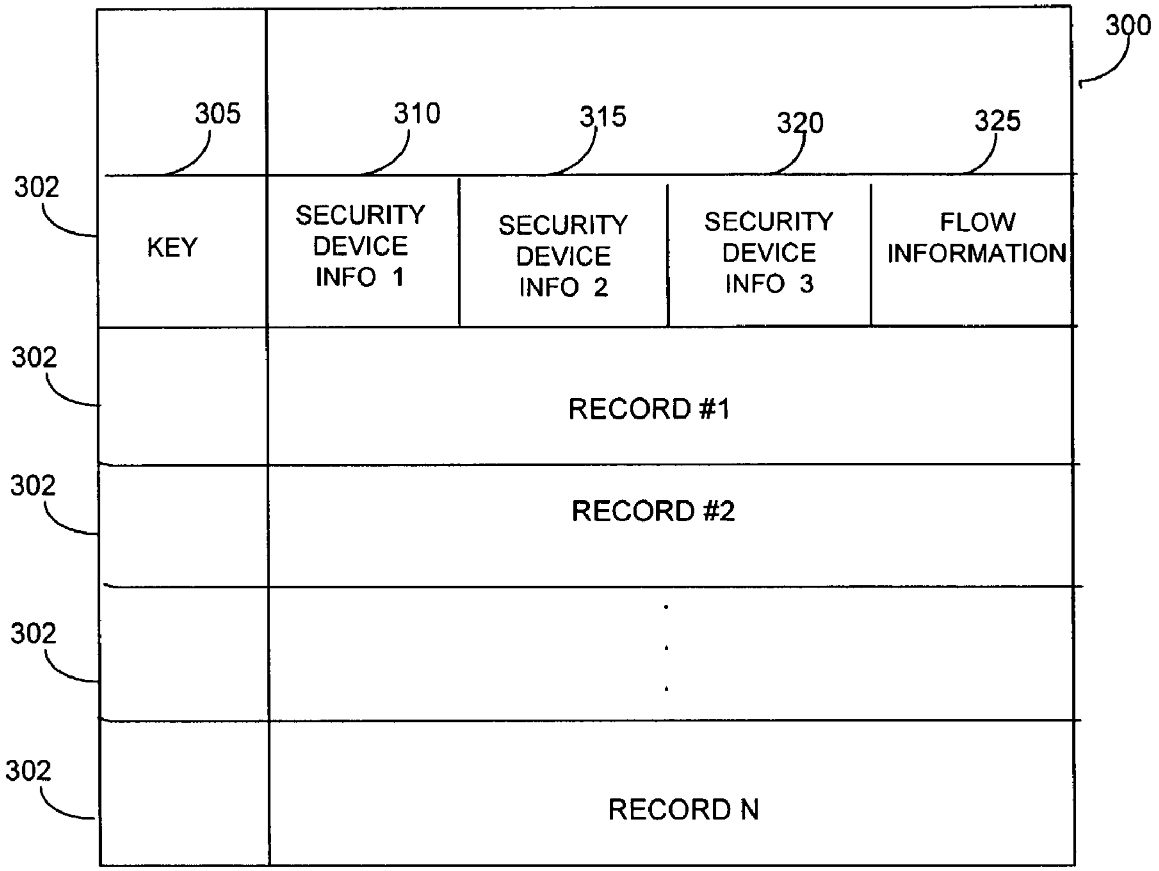

300
305
310
315
320
325
302
KEY
SECURITY DEVICE INFO 1
SECURITY DEVICE INFO 2
SECURITY DEVICE INFO 3
FLOW INFORMATION
302
RECORD #1
302
RECORD #2
302
302
RECORD N

FIG. 3

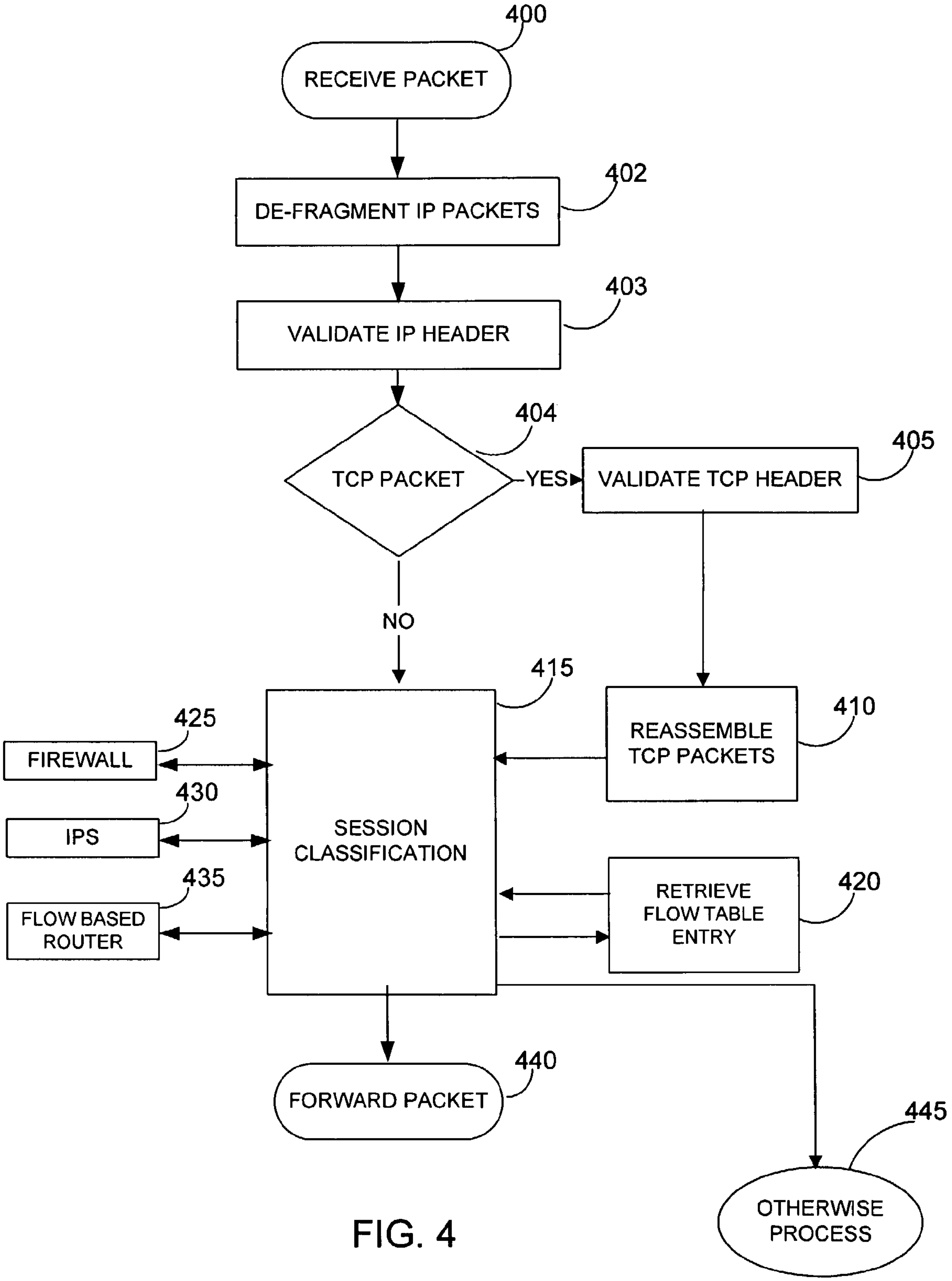

400
RECEIVE PACKET
402
DE-FRAGMENT IP PACKETS
403
VALIDATE IP HEADER
404
TCP PACKET
YES
405
VALIDATE TCP HEADER
NO
415
SESSION CLASSIFICATION
410
REASSEMBLE TCP PACKETS
425
FIREWALL
430
IPS
435
FLOW BASED ROUTER
420
RETRIEVE FLOW TABLE ENTRY
440
FORWARD PACKET
445
OTHERWISE PROCESS

FIG. 4

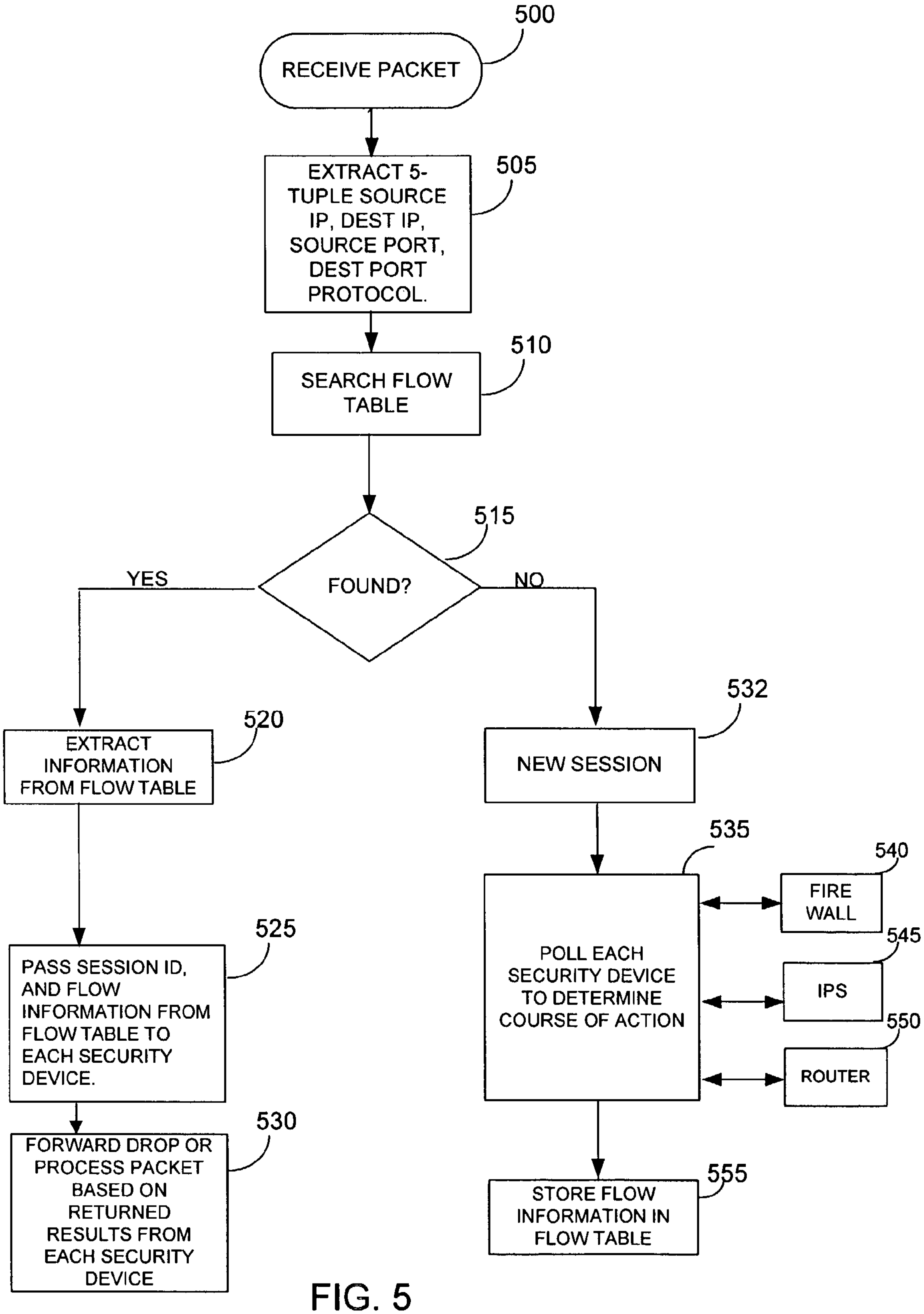

500
RECEIVE PACKET
505
EXTRACT 5-TUPLE SOURCE IP, DEST IP, SOURCE PORT, DEST PORT PROTOCOL.
510
SEARCH FLOW TABLE
515
FOUND?
YES
NO
520
EXTRACT INFORMATION FROM FLOW TABLE
525
PASS SESSION ID, AND FLOW INFORMATION FROM FLOW TABLE TO EACH SECURITY DEVICE.
530
FORWARD DROP OR PROCESS PACKET BASED ON RETURNED RESULTS FROM EACH SECURITY DEVICE
532
NEW SESSION
535
POLL EACH SECURITY DEVICE TO DETERMINE COURSE OF ACTION
540
FIRE WALL
545
IPS
550
ROUTER
555
STORE FLOW INFORMATION IN FLOW TABLE

FIG. 5

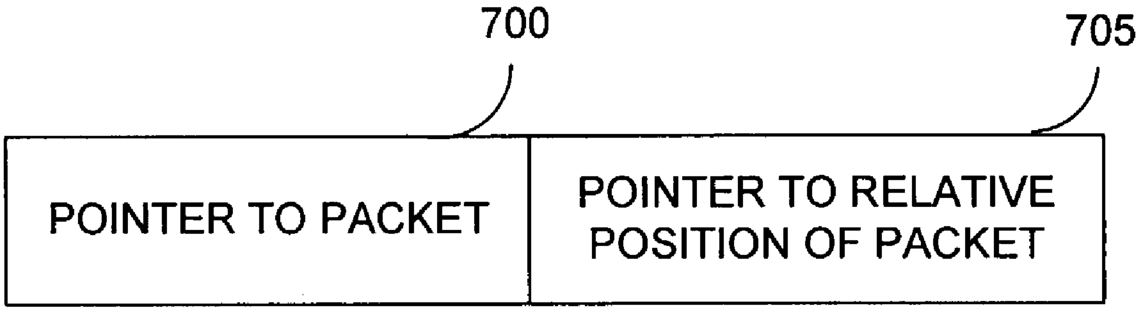

700
705
POINTER TO PACKET
POINTER TO RELATIVE POSITION OF PACKET

FIG. 6

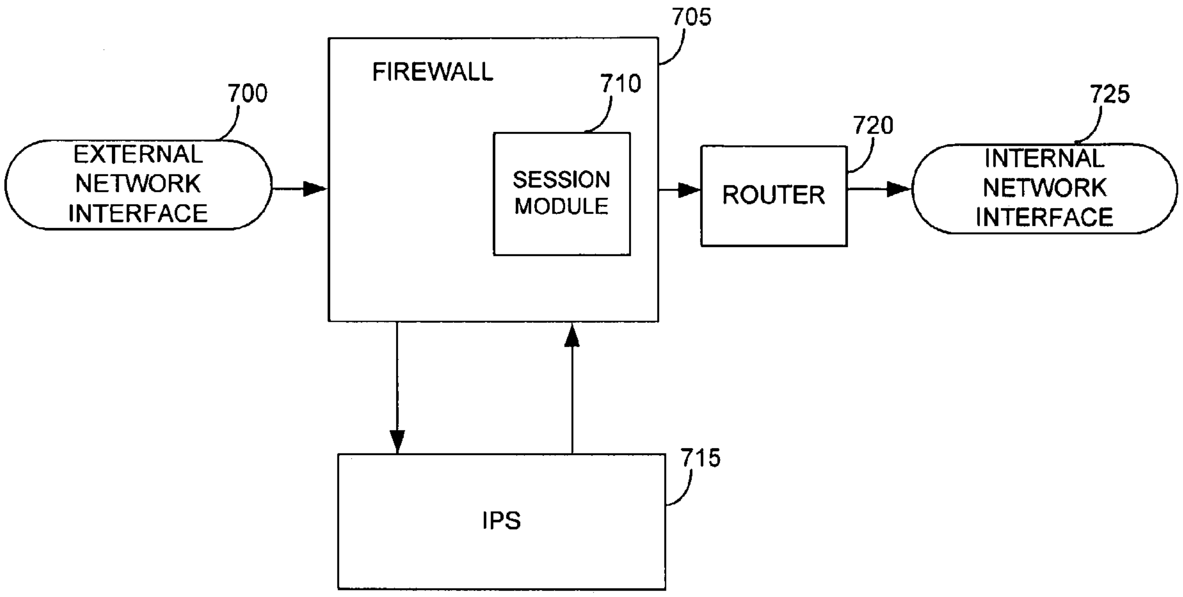

705
FIREWALL
710
SESSION MODULE
700
EXTERNAL NETWORK INTERFACE
720
ROUTER
725
INTERNAL NETWORK INTERFACE
715
IPS

FIG. 7

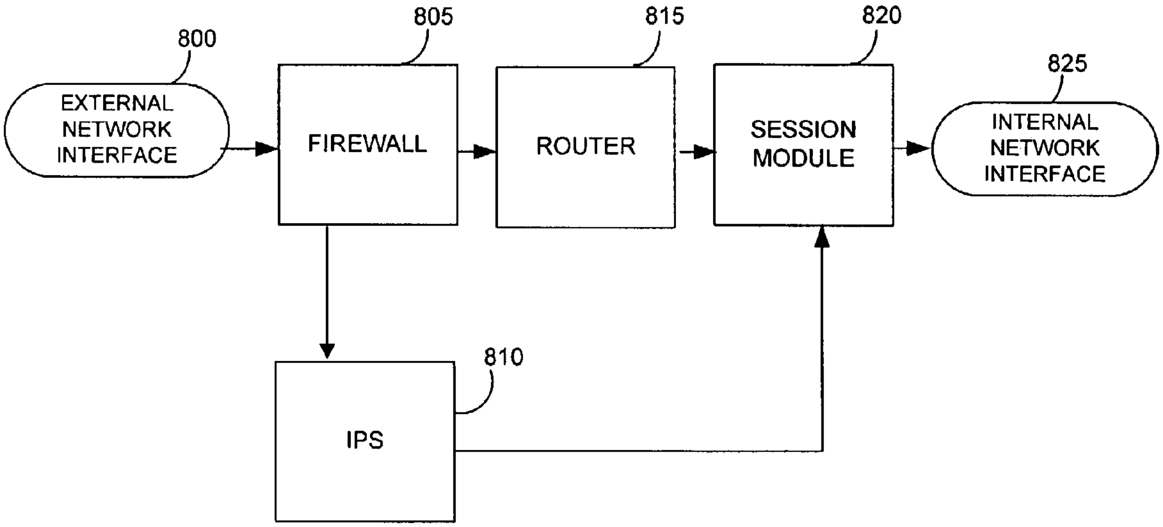

800
EXTERNAL NETWORK INTERFACE
805
FIREWALL
815
ROUTER
820
SESSION MODULE
825
INTERNAL NETWORK INTERFACE
810
IPS

FIG. 8

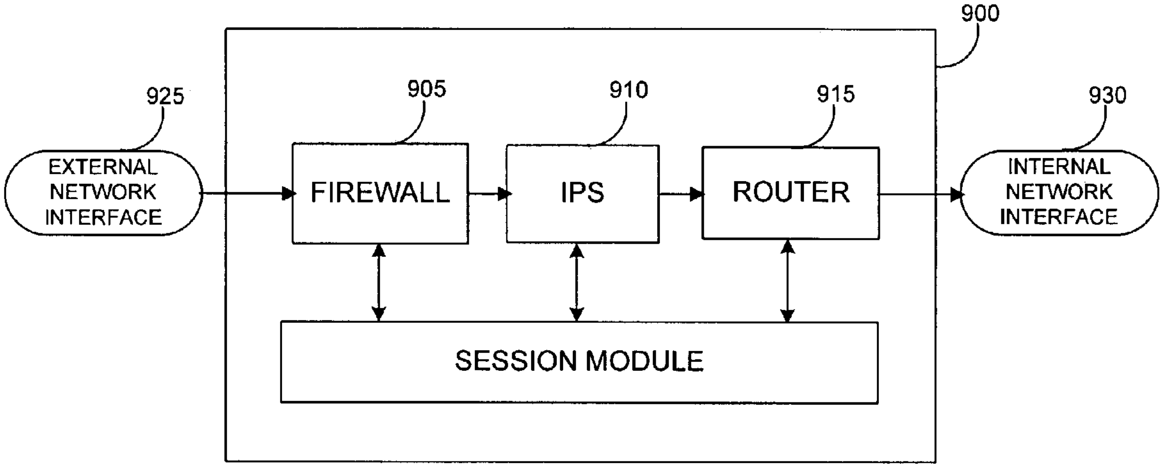

900
925
905
910
915
930
EXTERNAL NETWORK INTERFACE
FIREWALL
IPS
ROUTER
INTERNAL NETWORK INTERFACE
SESSION MODULE

FIG. 9

INTELLIGENT INTEGRATED NETWORK SECURITY DEVICE

CROSS-REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part (and claims the benefit of priority under 35 USC 120) of U.S. application Ser. No. 10/072,683, filed Feb. 8, 2002. The disclosure of the prior application is considered part of (and is incorporated by reference in) the disclosure of this application.

BACKGROUND

The present invention relates to a method for controlling computer network security.

Firewalls and intrusion detection systems are devices that are used to protect a computer network from unauthorized or disruptive users. A firewall can be used to secure a local area network from users outside the local area network. A firewall checks, routes, and frequently labels all messages sent to or from users outside the local area network. An intrusion detection system (IDS) can be used to examine information being communicated within a network to recognize suspicious patterns of behavior. Information obtained by the IDS can be used to block unauthorized or disruptive users from accessing the network. An intrusion prevention system (IPS) is an in-line version of an IDS. An IPS can be used to examine information as it is being communicated within a network to recognize suspicious patterns of behavior.

A flow-based router (FBR) can allow network administrators to implement packet forwarding and routing according to network policies defined by a network administrator. FBRs can allow network administrators to implement policies that selectively cause packets to be routed through specific paths in the network. FBRs can also be used to ensure that certain types of packets receive differentiated, preferential service as they are routed. Conventional routers can forward packets to their destination address based on available routing information. Instead of routing solely based on the destination address, FBRs can enable a network administrator to implement routing policies to allow or deny packets based on several other criteria including the application, the protocol, the packet size and the identity of the end system.

A packet filter can operate on the data in the network layer, to defend a trusted network from attack by an untrusted network. Packet filters can operate at the network layer to inspect fields of the TCP/IP header including, the protocol type, the source and destination IP address, and the source and destination port numbers. Disadvantages of packet filters include, speed (i.e., slow) and management in large networks with complex security policies. Packet filters alone may not provide robust protection because packet filters are not aware of the context of the given communication. In addition, packet filters do not inspect the data at the application layer making packet filters vulnerable to attempted security intrusions using the application layer.

A proxy server can operate on values carried in the application layer to insulate a trusted network from an untrusted network. In an application proxy server, two TCP connections are established: one between the packet source and the proxy server, another between the proxy server and the packet destination. The application proxy server can receive the arriving packets on behalf of the destination server. The application data can be assembled and examined by the proxy server, and a second TCP connection can be opened between the proxy server and the destination server to relay permitted packets to the destination server. Proxy servers can be slow because of the additional protocol stack overhead required to inspect packets at the application layer. Furthermore, because a unique proxy can be required for each application, proxy servers can be complex to implement and difficult to modify for supporting new applications. In addition, because proxy servers only examine application packets proxy servers may not detect an attempted network security intrusion at the TCP or network layers.

SUMMARY

The present invention provides methods and apparatus, including computer program products, for processing data packets and for implementing computer network security.

Advantages of the invention may include one or more of the following features. The technique disclosed can be used to detect an attempted network security intrusion and potentially block the current packet associated with the security intrusion. The disclosed technique can provide robust and efficient network security and includes plural security devices but only one flow table. Network security information is obtained from other network security devices and stored in a single flow record in the flow table. The use of a single flow record to determine whether a packet should be allowed can result in faster response time.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

DESCRIPTION OF DRAWINGS

FIG. **1** shows a network topology including a session module.

FIG. **2** illustrates a block diagram of the session module.

FIG. **3** shows the structure of a flow table.

FIG. **4** is a flowchart describing the operation of the session module.

FIG. **5** is a flowchart describing session classification.

FIG. **6** shows the quasi-reassembly information generated by the session module.

FIG. **7** shows a network topology where the session module is included in a firewall.

FIG. **8** shows a network topology where the session module operates in series with a firewall, IPS, and router.

FIG. **9** shows a network topology where a session module, firewall, IPS and router are included in a single security device.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

FIG. **1** shows a network topology including a local area network (LAN) (**100**), including a server (**102**), several workstations (W/S) (**104**), and a security device **124**. The security system **124** can include a session module **122** and a plurality of other security devices. In the implementation shown, the security system **124** includes two security devices, a first security device **106** and a second security device **108**. The LAN **100** is connected to an external network e.g., the Internet (**114**), through the security system **124**. The LAN **100** is also connected to a second LAN (**116**) through a router (**118**), and satellite (**120**). Second LAN **116** includes a web server (**110**), an email server (**112**), a server **102**, several workstations **104** and a security system **124**. The computers, servers and other devices in the LAN are interconnected using a number of data transmission media such as wire, fiber optics, and radio waves. The session module **122** monitors packets being communicated within the network. In one implementation, the first security device **106** can be a firewall and the second security device **108** can be an IPS. The session module **122** can act in conjunction with the first security device **106** and the second security device **108** to facilitate the blocking of packets associated with an attempted network security intrusion.

FIG. **2** shows a block diagram of a session module **122**. The session module **122** includes an incoming packet interface **205** for receiving packets. The received packets are analyzed by a flow processing engine (FPE) **202** to determine if an attempted network security intrusion is in progress. The session module **122** also includes a flow table **215**. The flow table **215** is used to store information regarding flows associated with received packets. The session module **122** also includes interfaces to other security devices on the network. In one implementation, the session module **122** includes a firewall interface **220**, an IPS interface **225**, and a flow router interface **230**. The security device interfaces **220** are used by the session module to obtain information regarding the received packet, and information regarding the flow associated with the packet, in order to determine if the received packet should be allowed or modified. The security device interfaces **218** are also used by the session module **122** to communicate flow information required by the security devices to facilitate processing of the packet.

FIG. **3** illustrates a structure of a flow table **300**. The flow table **300** includes flow records **302** associated with current TCP/IP flows. A TCP/IP flow includes a sequence of data packets communicating information between a source and a destination in one direction. The flow records are indexed using an indexing key **305**. The indexing key **305** is used to store and retrieve the appropriate flow record associated with a received packet. In one implementation, the indexing key **305** can be a hash key and the flow table **300** can be implemented as a hash table. The session module **122** (FIG. **2**) stores instructions for two or more security devices on the network in the same flow record. In one implementation of the session module **122**, instructions for three security devices (i.e. devices **310**, **315**, and **320**) are stored in the flow record **302**. The flow record **302** can store policy information (firewall policy, IPS policy etc., to apply to the flow) as well as other information that is used by the security devices such as encryption parameters, address translation parameters, bookkeeping information, and statistics. The flow record **302** can also include flow information **325** required by the session module **122** in order to decide whether the packet should be allowed. Such information can include information required to implement network policies regarding, for example connection time out, time billing, and bandwidth usage. Flows, sessions and flow tables are described in greater detail in co-pending and commonly owned patent application entitled “Multi-Method Gateway-Based Network Security Systems and Methods,” and assigned Ser. No. 10/072,683, the contents of which are expressly incorporated herein by reference.

FIG. **4** is a flow diagram describing the operation of the FPE **202** (FIG. **2**). Referring now to FIGS. **2** and **4**, incoming packets are received by the session module (step **400**). IP packets are de-fragmented (step **402**) and the IP header is validated for each IP packet (step **403**). In the validation step, the IP header associated with a given packet is extracted and the extracted IP header is inspected for fundamental flaws. Thereafter FPE **202** determines if the session is to be allowed (step **415**).

If the packet is a TCP packet (step **404**), the TCP header is validated (step **405**) and the TCP packets are reassembled (step **410**). The validation process includes extracting TCP header data and evaluating the header for fundamental flaws. The quasi-reassembly information developed in step **410** can be communicated by the session module **122** to other security devices to facilitate processing of the packet by the other security devices. Reassembly is described in greater detail below and in “Multi-Method Gateway-Based Network Security Systems and Methods.”

In step **415**, FPE **202** performs session classification using the TCP/IP header data associated with a given received packet. The session module **122** can determine if the packet should be allowed based on information obtained regarding the TCP/IP flow associated with the received packet and retrieved from the flow table **420**. In addition, the session module **122** can use information returned from one of the other security devices e.g., the firewall **425**, the IPS **430**, and the flow based router **435**. Further, the session module **122** can also facilitate the operation of the security devices by communicating flow information to a respective device as required by the device to process a given packet. Finally, FPE **202** forwards the packet if the packet should be allowed (step **440**). Otherwise, the packet is otherwise processed at step **445**. Other processing can include logging particular information regarding the packet, holding the packet, modifying and/or dropping the packet. This completes the description of the operation of FPE **202**.

FIG. **5** is a flow diagram showing the steps included in session classification (step **415**). The session classification step receives a packet (step **500**) and extracts information required to determine whether the packet should be allowed. The extracted information can include the source and destination IP addresses, the source and destination port numbers, and the protocol (step **505**). The extracted information can be used to search the flow table (step **510**) in order to determine if the packet is associated with a known session flow. For a known session flow, step **510** will produce a matching flow record in the flow table (step **515**). If a matching flow record is found, the FPE **202** (FIG. **2**) can extract TCP/IP session information for the received packet (step **520**) from the matching flow record. The FPE **202** determines if the received packet should be allowed using the TCP/IP session information obtained during step **520**. More specifically, the FPE **202** extracts information from the matching flow record, and passes the information to the security devices (e.g., communicating the session ID and the TCP/IP session information as well as any other security device specific information from the flow record) (step **525**). Depending on the returned results from the security devices, the FPE **202** can forward, drop, log, store, modify or otherwise process the given packet (step **530**).

If a matching flow record is not found in the flow table during step **515**, the received packet can be associated with a new TCP/IP session (step **532**). For a new TCP/IP session, the FPE **202** can assign a session ID to the new session and the FPE **202** can communicate with the other security devices (e.g. firewall, IPS, flow router) to determine a security policy for packets associated with the new session. For example, the FPE **202** can obtain information from the firewall **540** in order to determine if received packets associated with the new session should be allowed. The FPE **202** can communicate with the IPS **545** in order to determine if the received packet should be blocked because it matches known attack signatures for attempted network security intrusions. The FPE **202** can obtain any network policy associated with the new session from the flow router **550**. The FPE **202** can act as an arbiter between the different security devices and use the information obtained from the security devices either individually or in combination to determine if the packets associated with the new TCP/IP session should be allowed. The FPE **202** can use the information obtained from the security devices to create a new flow record and store the new flow record in the flow table (step **555**). The new flow record includes the TCP/IP session information for the new session associated with the received packet and any other specific security device information. Thereafter, the FPE **202** can facilitate the processing of received packets associated with a given TCP/IP session as described above in association with FIG. **4** including communicating the session ID, TCP/IP session information and security device specific information to the security devices from a corresponding flow record.

In addition to determining if a received packet is associated with an attempted network security intrusion using the varied security devices, the session module can also perform quasi-reassembly of the received TCP/IP packets as described above in association with FIG. **4**. FIG. **6** shows the quasi-reassembly information that can be generated by the session module. The quasi-reassembly information can include a pointer to a location of a given packet **600** in memory and a pointer to information containing the relative position of the packet in a flow **605**. In one implementation, an IPS can perform passive TCP/IP reassembly and the pointer to the location of the packet can be used to locate the packet within the IPS. In another implementation, the pointer to information containing the relative position of the packet in the flow can be used to obtain the TCP/IP sequence number included in the TCP/IP header associated with the packet. The quasi-reassembly information can be communicated to the security devices connected to the session module **122** (FIG. **2**) as required. The security devices can use the quasi-reassembly information to process the received packet.

The session module can be used in a number of different network topologies. FIG. **7** shows a network topology where a session module **710** is integrated into a firewall **705**. The firewall **705** can include an interface to a router **720** and an IPS **715**. The firewall **705** receives packets from the external network interface **700**. The firewall **705** communicates with the IPS **715** to determine whether the received packet should be blocked based on known attack signatures. If the firewall **705** and IPS **715** determine that the packet should be allowed to pass, the firewall **705** sends the received packet to the router **720**. The router **720** forwards the outgoing packet to its intended destination, using the internal network interface **725**, based on the network policies stored in the router.

FIG. **8** shows an alternate arrangement for implementing computer network security using a session module. In this arrangement, the session module **820** operates in series with a firewall **805**, an IPS **810**, and a router **815**. Packets received using the external network interface **800** are screened by the firewall **805** before being communicated to the router **815**. The firewall **805** also sends information regarding the received packet to the IPS **810**. The IPS **810** examines the received packet and informs the session module **820** if the received packet should be blocked based on known attack signatures. The router **815** sends the packet to the session module **820** for further processing. If the session module **820** determines that the received packet should be allowed it forwards the received packet to its intended destination using the internal network interface **825**.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Method steps of the invention can be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method steps can also be performed by, and apparatus of the invention can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of nonvolatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

This invention has been described in terms of particular embodiments. Nevertheless, it will be understood that various modifications may be made without departing with the spirit and scope of the invention. For instance, the steps of the invention can be performed in a different order and still achieve desirable results. In addition, the session module, IPS, firewall, and router can all be incorporated into a single device such as the configuration shown in FIG. **9**. Other configurations of a session module packaged with one or more security devices are also possible. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A method for inspecting data packets associated with a flow in a computer network, the computer network including two or more security devices for processing the data packets, each data packet having associated header data, the method comprising:

receiving the data packet;

examining the data packet;

determining a single flow record associated with the data packet, where the determining includes:

determining a packet identifier using at least the associated header data;

evaluating a flow table for a matching flow record entry using the packet identifier;

when there is a matching flow record entry, retrieving the matching flow record;

when there is no matching flow record entry, creating a new flow record; and

storing the new flow record in the flow table;

extracting flow instructions, a session ID and flow information, for the two or more security devices, from the single flow record and forwarding the flow instructions, the session ID and the flow information to the respective ones of the two or more security devices to facilitate processing of the data packet;

receiving, from each of the two or more security devices, evaluation information, the evaluation information being generated by a respective one of the two or more security devices when processing the data packet; and

processing the data packet using the evaluation information.

2. The method of claim **1**, where the two or more security devices are included in one of an intrusion detection system, an intrusion prevention system, a firewall, or a flow-based router.

3. The method of claim **1**, where examining the data packet includes inspecting the associated header data to determine if the data packet should be forwarded.

4. The method of claim **1**, where determining the single flow record associated with the data packet includes locating a flow record using the associated header data.

5. The method of claim **1**, where the extracting flow instructions for the two or more security devices from the single flow record includes obtaining information for locating the data packet in memory and information providing a relative position of the data packet within the flow.

6. The method of claim **1**, where creating the new flow record includes:

creating a new session ID;

retrieving device specific flow information associated with the data packet from the two or more security devices; and

associating the new session ID and the device specific flow information with the new flow record.

7. The method of claim **6**, further comprising using the device specific flow information to create instructions for each of the two or more security devices for processing the packet.

8. The method of claim **6**, further comprising storing security device specific flow information for each of the two or more security devices along with the new session ID in the single flow record.

9. The method of claim **1**, further comprising processing the data packet in each of the two or more security devices using the extracted flow instructions.

10. The method of claim **1**, where processing the data packet includes one or more of forwarding, dropping, modifying, logging or storing the data packet.

11. The method of claim **1**, where processing the data packet includes determining if the data packet is to be forwarded.

12. The method of claim **1**, where the single flow record includes policy information for at least one of the two or more security devices.

13. The method of claim **1**, where the single flow record includes a firewall policy and an intrusion prevention system policy.

14. The method of claim **1**, where the single flow record includes encryption parameters for use by one of the two or more security devices.

15. The method of claim **1**, where the single flow record includes address translation parameters.

16. The method of claim **1**, where the single flow record includes bookkeeping or statistics information.

17. The method of claim **1**, where the single flow record includes information describing which policies to apply to a given flow for use by one or more of the two or more security devices.

18. The method of claim **1**, where the single flow record includes policy information for use by the two or more security devices in processing the data packet.

19. A computer-readable memory device incorporating instructions for inspecting data packets associated with a flow in a computer network, the computer network including two or more security devices for processing data packets, each data packet having associated header data, the instructions to:

receive the data packet;

examine the data packet;

determine a single flow record associated with the data packet, where the instruction to determine the single flow packet include instructions to:

determine a packet identifier using at least the associated header data;

evaluate a flow table for a matching flow record entry using the packet identifier;

retrieve a matching flow record when there is a matching flow record entry; and

create a new flow record when there is no matching flow record entry, where the new flow record is stored in the flow record table;

extract flow instructions, a session ID and flow information, for the two or more security devices, from the single flow record and forward the flow instructions, the session ID and the flow information to the respective ones of the two or more security devices to facilitate processing of the data packet;

receive, from each of the two or more security devices, evaluation information, the evaluation information being generated by a respective one of the two or more security devices when processing the data packet; and

processing the data packet using the evaluation information.

20. The computer-readable memory device of claim **19**, where the two or more security devices are included in one of an intrusion detection system, an intrusion prevention system, a firewall or a flow-based router.

21. The computer-readable memory device of claim **19**, where instructions to examine the data packet further include instructions to inspect the associated header data to determine if the data packet should be forwarded.

22. The computer-readable memory device of claim **19**, where instructions to determine the single flow record associated with the data packet further include instructions to locate a flow record using the associated header data.

23. The computer-readable memory device of claim **19**, where instructions to extract flow instructions for the two or more security devices from the single flow record further include instructions to obtain information for locating the data packet in memory and information providing a relative position of the data packet within the flow.

24. The computer-readable memory device of claim **19**, where instructions to create the new flow record include instructions to:

create a new session ID;

retrieve device specific flow information associated with the data packet from the two or more security devices; and

associate the new session ID and the device specific flow information with the new flow record.

25. The computer-readable memory device of claim **24**, further comprising instructions to use the device specific flow information to create instructions for each of the two or more security devices for processing the data packet.

26. The computer-readable memory device of claim **24**, the computer-readable memory device further comprising instructions to store security device specific flow information for each of the two or more security devices along with the new session ID in the single flow record.

27. The computer-readable memory device of claim **19**, further comprising instructions to process the data packet in each of the two or more security devices using the extracted flow instructions.

28. The computer-readable memory device of claim **19**, where instructions to process the data packet include one or more instructions to forward, drop, log or store the data packet.

29. The computer-readable memory device of claim **19**, where instructions to process the data packet include instructions to determine if the data packet is to be forwarded.

30. The computer-readable memory device of claim **19**, where the single flow record includes policy information for at least one of the two or more security devices.

31. The computer-readable memory device of claim **19**, where the single flow record includes a firewall policy and an intrusion prevention system policy.

32. The computer-readable memory device of claim **19**, where the single flow record includes encryption parameters for use by the two or more security devices.

33. The computer-readable memory device of claim **19**, where the single flow record includes address translation parameters.

34. The computer-readable memory device of claim **19**, where the single flow record includes bookkeeping or statistics information.

35. The computer-readable memory device of claim **19**, where the single flow record includes information describing which policies to apply to a given flow for use by one or more of the two or more security devices.

36. The computer-readable memory device of claim **19**, where the single flow record includes policy information for use by two or more security devices in processing the data packet.

37. An apparatus for processing data packets, having associated header data, comprising:

a session module to determine flow information for each received data packet and evaluate a packet identifier, associated with the header data, identifying a particular flow associated with a given data packet;

a flow table that includes flow records for each flow having information determined by the session module, each flow record including flow information for a plurality of security devices coupled to the apparatus,

where the session module is further to:

to locate a flow record, in the flow table, associated with the identified particular flow and retrieve the located flow record;

transmit device specific flow information, including flow instructions associated with the located flow record, a session ID associated with the located flow record and flow information associated with the located flow record, to each of the plurality of security devices via communication interfaces;

receive, from each of the plurality of security devices, evaluation information, the evaluation information being generated by a respective one of the plurality of security devices in processing the data packets; and

process the data packets using the evaluation information.

38. The apparatus of claim **37**, where the session module is further to process the given data packet by one of dropping, logging, storing, or forwarding the given data packet.

39. The apparatus of claim **37**, where the flow table includes an index key and the device specific flow information for the plurality of security devices.

40. The apparatus of claim **37**, where the plurality of security devices are included in one of a firewall, a flow-based router, an intrusion detection system, or an intrusion prevention system.

41. The apparatus of claim **37**, where the flow table includes one or more flow records that include policy information for use by the plurality of security devices in processing the data packets.

* * * * *

EXHIBIT D

US007302700B2

(12) United States Patent

Mao et al.

(10) Patent No.: US 7,302,700 B2

(45) Date of Patent: Nov. 27, 2007

(54) **METHOD AND APPARATUS FOR IMPLEMENTING A LAYER 3/LAYER 7 FIREWALL IN AN L2 DEVICE**

(75) Inventors: **Yu Ming Mao**, Milpitas, CA (US); **Roger Jia-Jyi Lian**, San Jose, CA (US); **Guangsong Huang**, Sunnyvale, CA (US); **Lee Chik Cheung**, San Jose, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 964 days.

(21) Appl. No.: **09/967,878**

(22) Filed: **Sep. 28, 2001**

(65) **Prior Publication Data**

US 2003/0065944 A1 Apr. 3, 2003

(51) **Int. Cl.**
G06F 17/00 (2006.01)
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **726/11**; 713/150

(58) **Field of Classification Search** 709/220–230; 713/200–205, 150; 370/400–401; 726/13, 726/14, 11
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,544,322	A	*	8/1996	Cheng et al.	709/229
5,708,654	A	*	1/1998	Arndt et al.	370/242
5,889,953	A	*	3/1999	Thebaut et al.	709/221
5,905,859	A	*	5/1999	Holloway et al.	713/201
5,918,018	A	*	6/1999	Gooderum et al.	709/225
5,968,176	A	*	10/1999	Nessett et al.	713/201
6,000,045	A	*	12/1999	Lewis	714/47
6,041,058	A	*	3/2000	Flanders et al.	370/401
6,115,472	A		9/2000	Shimizu	
6,131,120	A		10/2000	Reid	
6,141,755	A	*	10/2000	Dowd et al.	726/11
6,182,226	B1	*	1/2001	Reid et al.	713/201
6,212,558	B1	*	4/2001	Antur et al.	709/221
6,219,707	B1	*	4/2001	Gooderum et al.	709/225
6,233,688	B1		5/2001	Montenegro	
6,304,973	B1	*	10/2001	Williams	713/201
6,684,253	B1	*	1/2004	Whitaker et al.	709/229
6,754,716	B1	*	6/2004	Sharma et al.	709/238
6,763,469	B1	*	7/2004	Daniely	726/11
6,961,771	B2	*	11/2005	Sato	709/225
7,047,561	B1	*	5/2006	Lee	726/12
7,103,055	B2	*	9/2006	Kadambi et al.	370/409
2001/0042213	A1	*	11/2001	Jemes et al.	713/201

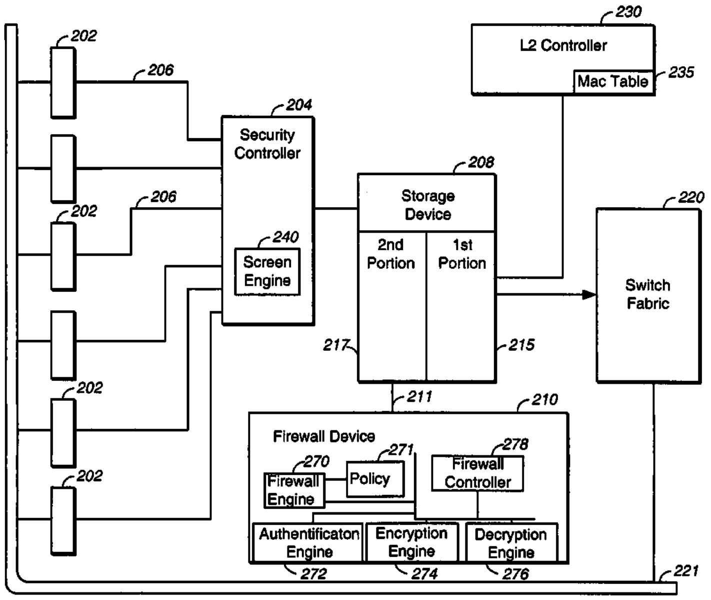
* cited by examiner

Primary Examiner—Christopher Revak
Assistant Examiner—Christian LaForgia
(74) *Attorney, Agent, or Firm*—Harrity Snyder, LLP

(57) **ABSTRACT**

Methods and apparatus for transferring packets in a packet switched communication system. A system is provided that includes an L2 device including a controller determining for each packet received whether the received packet is to be inspected, an inspection device operable to inspect and filter packets identified by the controller including using a zone specific policy and an L2 controller for transferring inspected packets in accordance with L2 header information using L2 protocols.

24 Claims, 5 Drawing Sheets



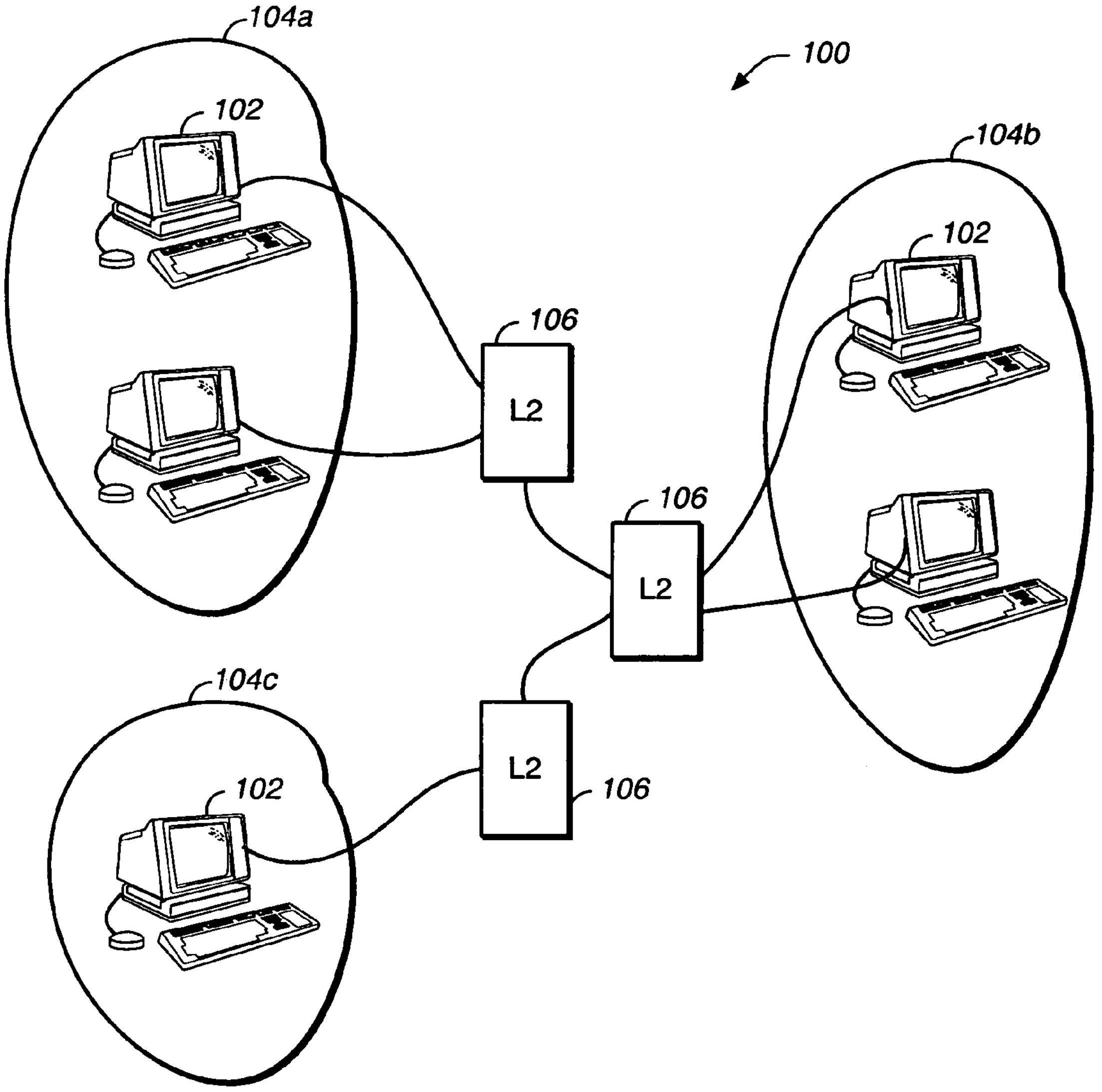


FIG._1

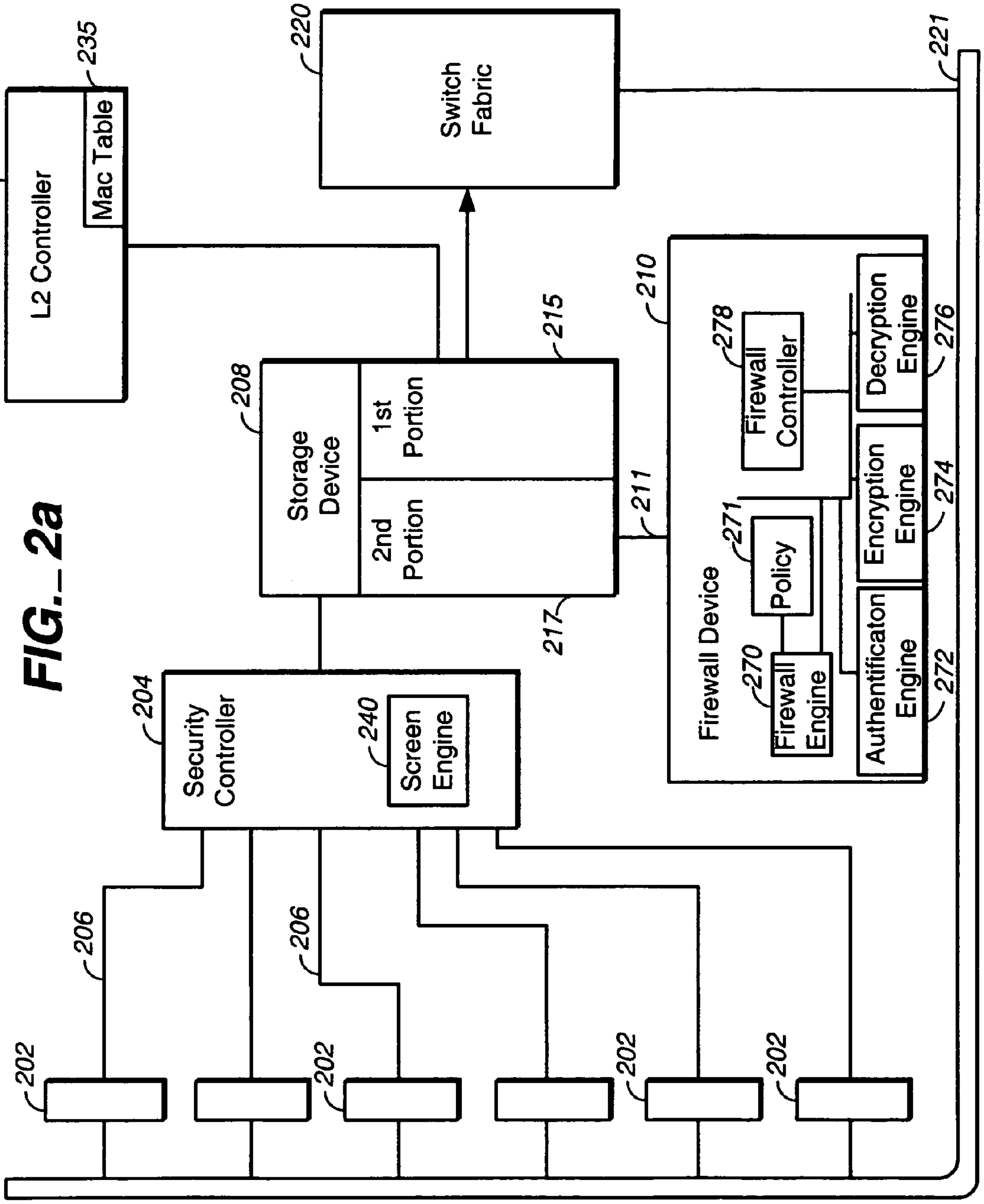

230
L2 Controller
235
Mac Table
220
Switch Fabric
221
208
Storage Device
1st Portion
2nd Portion
215
217
211
210
Firewall Device
278
Firewall Controller
271
Policy
270
Firewall Engine
Decryption Engine
276
Encryption Engine
274
Authentificaton Engine
272
204
Security Controller
240
Screen Engine
206
206
202
202
202
202

FIG._ 2a

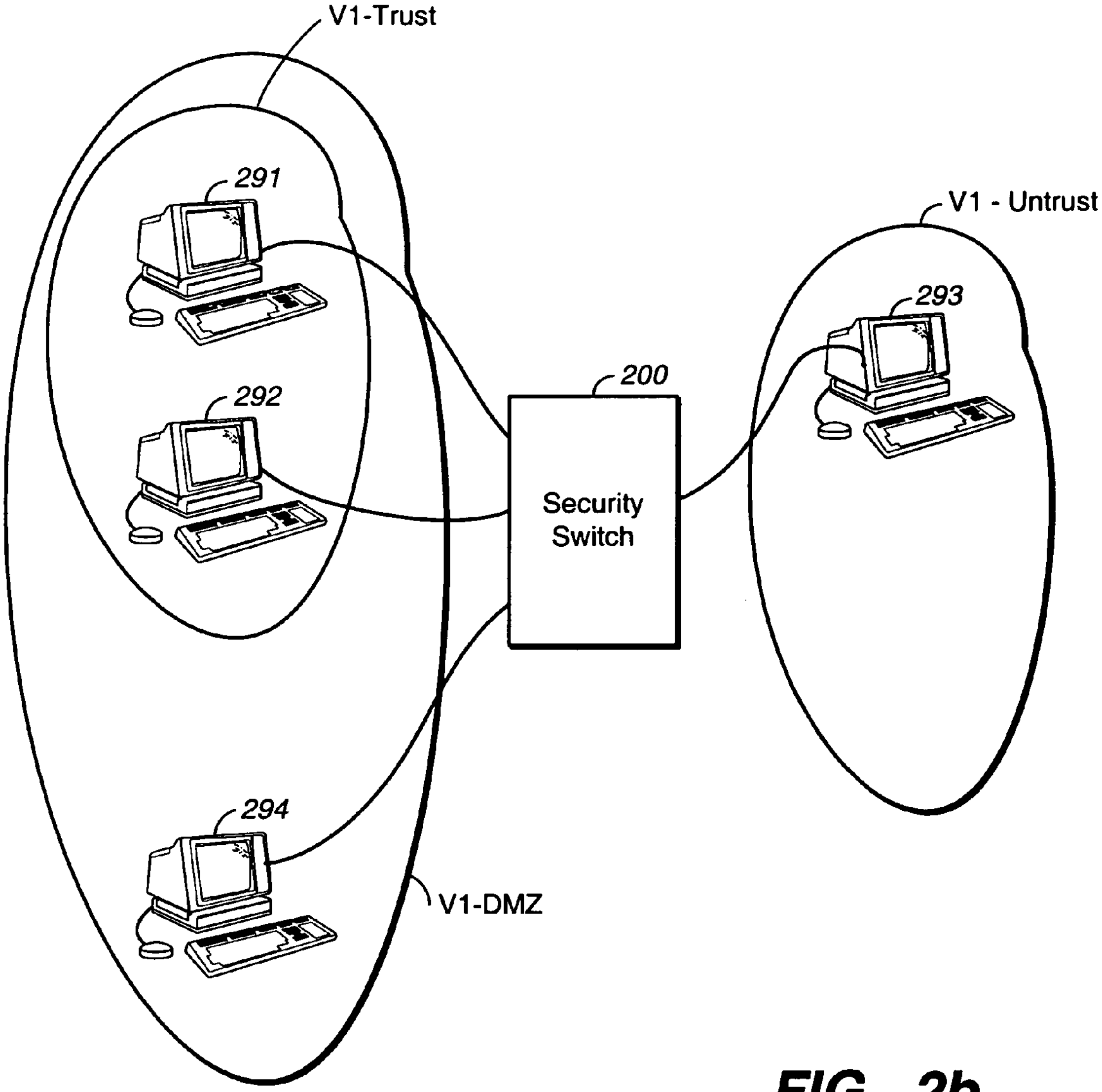

V1-Trust
291
292
294
V1-DMZ
200
Security Switch
V1 - Untrust
293

FIG._2b

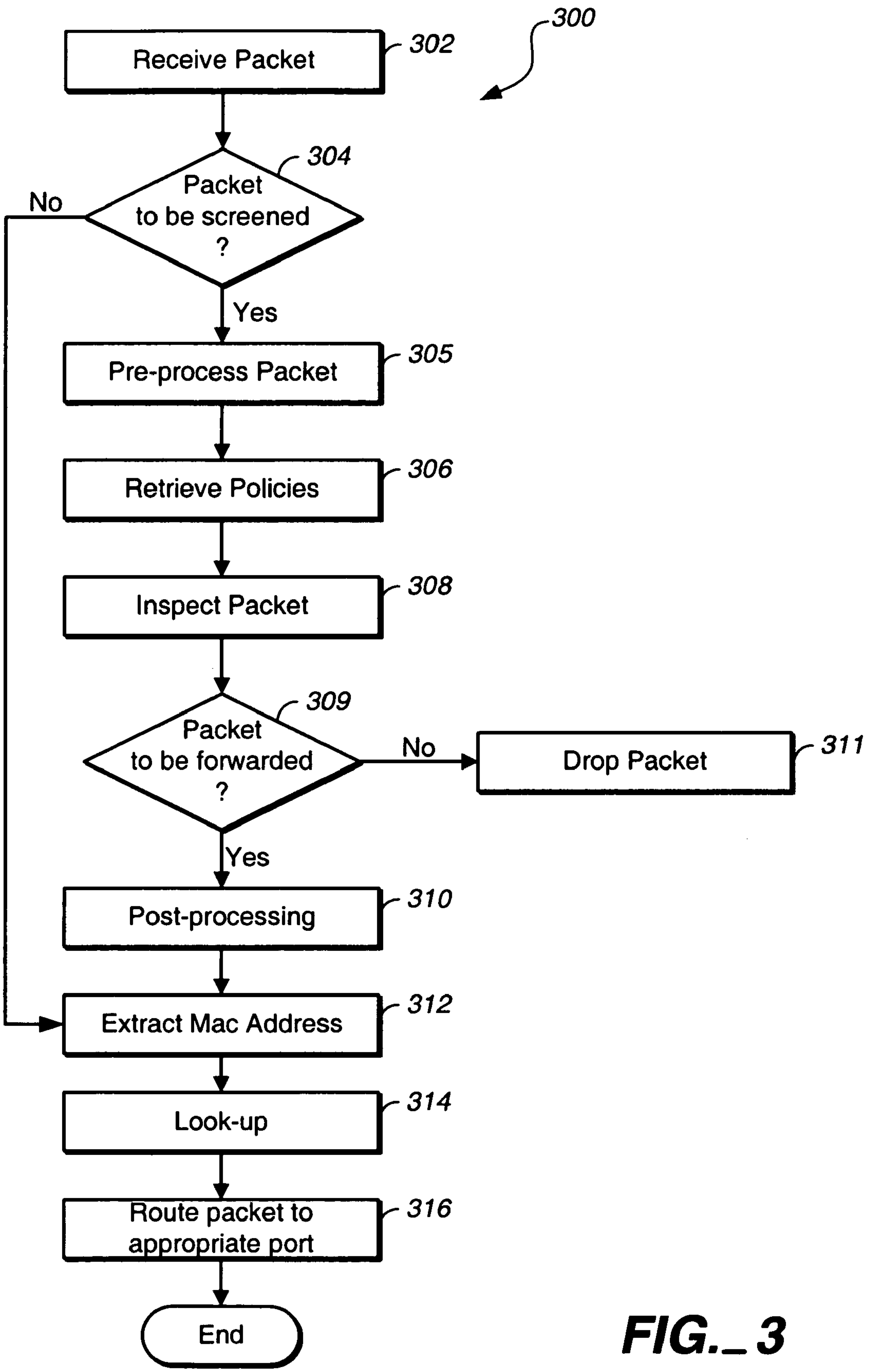

300
Receive Packet
302
Packet to be screened ?
304
No
Yes
Pre-process Packet
305
Retrieve Policies
306
Inspect Packet
308
Packet to be forwarded ?
309
No
Drop Packet
311
Yes
Post-processing
310
Extract Mac Address
312
Look-up
314
Route packet to appropriate port
316
End

FIG._3

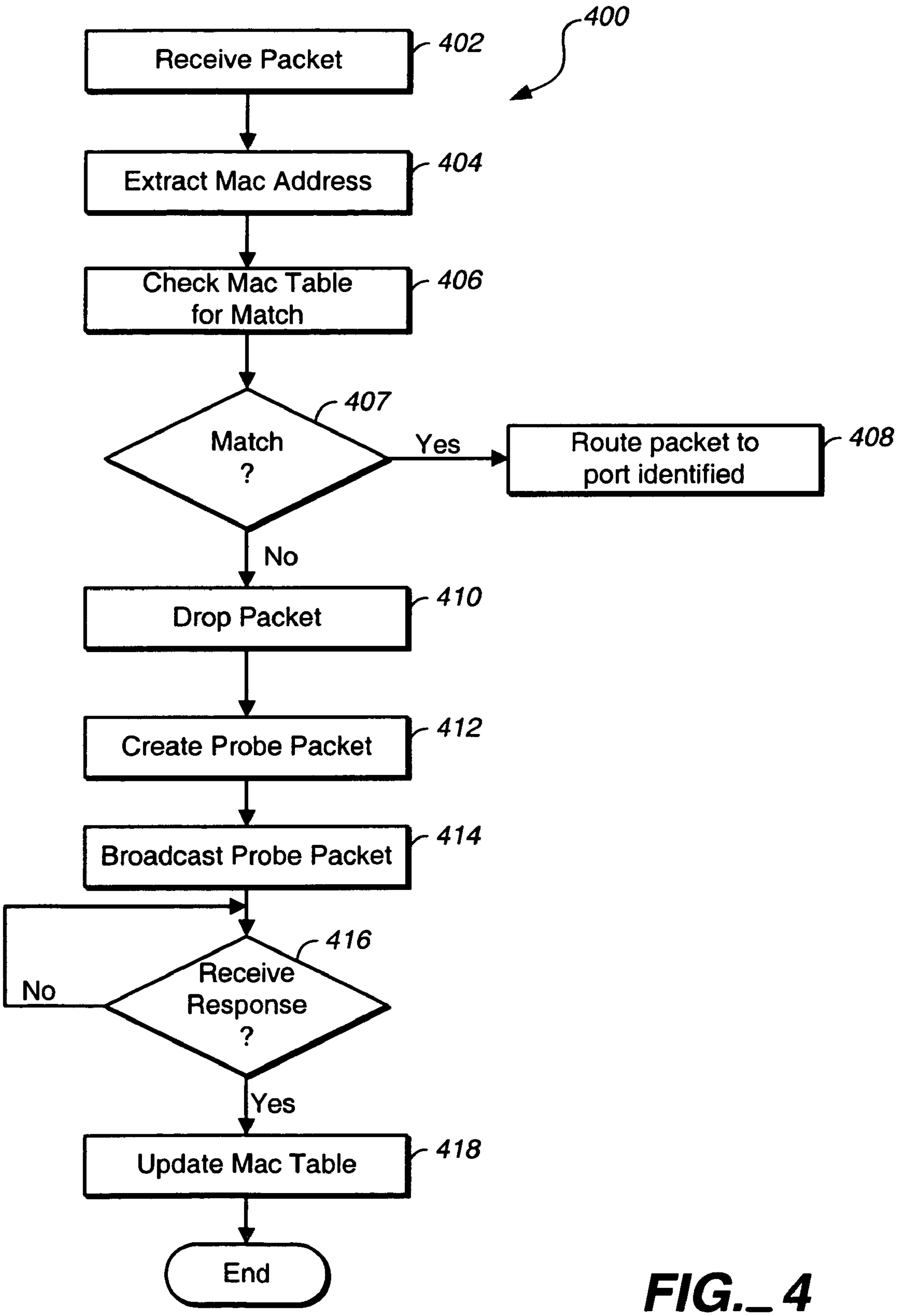

400
Receive Packet
402
Extract Mac Address
404
Check Mac Table for Match
406
Match ?
407
Yes
Route packet to port identified
408
No
Drop Packet
410
Create Probe Packet
412
Broadcast Probe Packet
414
Receive Response ?
416
No
Yes
Update Mac Table
418
End

FIG._4

METHOD AND APPARATUS FOR IMPLEMENTING A LAYER 3/LAYER 7 FIREWALL IN AN L2 DEVICE

BACKGROUND

The present invention relates generally to data routing systems, and more particularly to methods and apparatus for providing secure communications on a network.

A packet switch communication system includes a network of one or more switches or routers connecting a plurality of users. A packet is the fundamental unit of transfer in the packet switch communication system. A user can be an individual user terminal or another network.

A layer **2** (L2) switch is a switching device which receives packets containing data or control information on one port, and based on a media access connection (MAC) address contained within the packet, switches the packet out another port. Conventional L2 switches perform this switching function by evaluating layer **2** (L2) header information contained within the packet in order to determine the proper output port for a particular packet. The L2 switch includes a table that maps MAC addresses with output ports. If a MAC address is unknown (i.e., there is no corresponding entry in the table), then the corresponding packet is broadcast to all output ports with the hope that another component in the packet switched communication system will recognize the MAC address (and pass back information to the forwarding L2 switch to update its table). Other types of L2 devices include bridges.

A router is a switching device which receives packets containing data or control information on one port, and based on destination information contained within the packet, routes the packet to a next hop to/toward the destination. Conventional routers perform this switching function by evaluating layer **3** (L3) header information contained within the packet in order to determine a next hop for a particular packet. The layer **3** information includes an IP address associated with the intended destination (as well as source address) for the packet.

The network coupling the users can be an intranet, that is, a network connecting one or more private servers such as a local area network (LAN). Alternatively, the network can be a public network, such as the Internet, in which data packets are passed over untrusted communication links. The network configuration can include a combination of public and private networks. For example, two or more LAN's with individual terminals can be coupled together using a public network such as the Internet. Data security issues can arise when public and private networks are linked or when distinct networks are coupled. For example, conventional packet switched communication systems that include links between public and private networks typically include security measures for assuring network access control and data integrity.

In order to assure individual packet security, packet switched communication systems can include encryption/decryption services. Prior to leaving a trusted network (or portion of a network), individual packets can be encrypted to minimize the possibility of data loss while the packet is transferred over an untrusted (e.g., public) network (or portion thereof). Upon receipt at a destination or another trusted portion of the communication system (e.g., at a firewall just before the destination), the packet can be decrypted and subsequently delivered to its intended destination. The use of encryption and decryption allows for the creation of a virtual private network (VPN) between users separated by untrusted communication links.

In addition to security concerns for the data transferred over the public portion of the communications system, the private portions of the network must safeguard against intrusions through the gateway provided at the interface of the private and the public networks. A firewall is a device that can be coupled in-line between a public network and private network for screening packets received from the public network. A firewall is a particular type of L3/L4 device that can be used to enforce policy and filtering functions. A firewall can include one or more engines for inspecting, filtering, authenticating, encrypting, decrypting and otherwise manipulating received packets. Conventional firewalls use L3 and L4 header information including IP addresses associated with the source and destination of a given packet being processed. Received packets are inspected and thereafter forwarded or dropped in accordance with the policies associated with the given domain.

SUMMARY

In one aspect, the invention provides an L2 device in a packet switched communication system. The packet switched communication system has plural zones and each zone represents a distinct security domain and has an associated policy for use in inspecting packets entering/exiting an associated zone. The L2 device includes at least one port coupled to a terminal unit included in a first security zone, at least one port coupled to a terminal unit included in a second security zone, a controller determining for each packet received whether the received packet is destined for another zone, a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy and an L2 switching engine. The L2 switching engine is operable to immediately route to a port all intra-zone packets passing through the L2 device using a table of MAC addresses and corresponding ports, and only route to a port inter-zone packets that are retained after the inspection by the firewall engine.

In another aspect, the invention provides an L2 device in a packet switched communication system. The L2 device includes a controller determining for each packet received whether the received packet is to be transferred intra-zone or inter-zone, a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy and an L2 switching engine operable to immediately route to a port all intra-zone packets passing through the L2 device using a table of MAC addresses and corresponding ports and only route to a port inter-zone packets that are retained after the inspection by the firewall engine.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be transferred inter-zone and a firewall engine operable to inspect and filter inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be transferred inter-zone and an inspection device operable to inspect and filter inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides an L2 device in a packet switched communication system including a controller determining for each packet received whether the received packet is to be inspected, an inspection device operable to inspect and filter packets identified by the controller including using a zone specific policy and an L2 controller for transferring inspected packets in accordance with L2 header information using L2 protocols.

Aspects of the invention can include one or more of the following features. The inspection device can be a firewall including a layer **3** firewall device, a layer **4** firewall device and a layer **7** firewall device. The inspection device can be a firewall that filters based on layer information other than layer **2** header information. The controller can determine each packet that is to pass between security zones and the inspection device only processes inter-zone traffic. The controller can determine each packet that is to remain in a single security zone and the inspection device immediately routes intra-zone packets. The device can route traffic using the MAC address in the layer **2** header of a given packet to determine an egress port on the device to which the packet is to be routed. The device can include a storage element for storing packets that are to be inspected and an L2 controller for transferring packets through the device including determining an egress port for transferring a given packet using a destination MAC address in the given packet and a MAC address table that includes a mapping of MAC addresses and associated egress nodes. The memory element can include a first and second portion. The first portion can store packets to be transferred through the device and the second portion can store packets waiting for inspection. The device can be a L2 switch or an L2 bridge.

In another aspect, the invention provides a method for transferring packets in a communication network including receiving a packet at an L2 device, determining whether the received packet is to be transferred inter-zone and inspecting and filtering inter-zone packets using a zone specific policy prior to routing using L2 protocols.

In another aspect, the invention provides a method for transferring packets in a communication network including receiving a packet at an L2 device, determining whether the received packet is to be inspected and inspecting and filtering identified packets using a zone specific policy prior to transferring the packet through the L2 device using L2 protocols.

In another aspect, the invention provides a method for switching packets in a communication network including receiving a packet at an interface of an L2 device, determining if a destination MAC address associated with the received packet is known and, if not, holding the received packet a predetermined amount of time without transferring the packet to any port of the L2 device, creating a probe packet that includes the unknown MAC address and broadcasting the probe packet to all interfaces except the receiving interface.

Aspects of the invention can include one or more of the following features. The probe packet can include a time to life (TTL) field in a IP header and the method can include setting a value of the TTL field such that a downstream node having the unknown MAC address and receiving the probe cell will return an expired message to the L2 device. The method can include dropping the packet after the expiration of the predetermined amount of time. The packet can be dropped if the MAC address is unknown. The method can include receiving a response from on one of the broadcast interfaces and updating a table indicating a previously unknown MAC address is associated with the responding interface.

In another aspect, the invention provides method of providing secure communications between users without requiring encryption and decryption services at a respective user. The method includes identifying first and second users, coupling the first and second users through two or more L2 devices over a communication network and specifying a virtual private network for communications between the first and second users. The virtual private network is defined between a first and second L2 device in the network. The method includes receiving a packet at either the first or the second L2 device, determining whether the received packet is associated with the virtual private network and encrypting and decrypting as appropriate identified packets using local encryption and decryption services prior to transferring the packet through the L2 device using L2 protocols.

Aspects of the invention can include one or more of the following features. The step of determining can include using a destination MAC address associated with the packet to identify a virtual private network.

In another aspect, the invention provides a virtual private network for providing secure communications between users without requiring encryption and decryption services at a respective user. The virtual private network includes first and second L2 devices coupling first and second users over a communication network where each of the first and second L2 devices includes a screening mechanism determining whether a received packet is associated with the virtual private network and encryption and decryption services operating on packets associated with the virtual private network prior to a transfer of the packet through the L2 device using L2 protocols.

Aspects of the invention can include one or more of the following advantages. A packet switched communication system is provided that allows for the creation of plural security zones within a single device without requiring changes to the network or terminal configuration. Between each zone, a terminal unit can communicate with other terminal units without the knowledge of, yet receiving the benefits of, L2 switching and up to layer **7** security filtering as discussed below. A packet switched communication system is provided that includes L2 switch and firewall functionality. The packet switched communication system acts as an IEEE 802.1Q VLAN L2 conventional switch forwarding/filtering based on MAC-address for all intra-zone communications. The packet switched communication system allows L2 switching among multiple ports inside a given security zone. The L2 switch also provides up to layer **7** security firewall protections as appropriate for inter-zone or intra-zone traffic including TCP stateful inspection, syn-attack guard, policy-based control, load balancing and other functionalities on each data stream. In one implementation, the packet switched communication system can be configured to include multiple IEEE 802.1Q VLAN based L2 transparent domains. A user can create multiple VLANs, each having its own policy for firewall control. In addition, methods are provided for VPN tunnel capability to connect remote clients to the L2 domain. Methods are provided to guard against broadcasting information throughout the zones and violating one or more security constraints when a MAC address that is being processed is not recognized. The methods include the broadcast of probe packets to discover topology information for unknown MAC destinations.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. **1** is a block diagram of a packet switched communication system including an L2 firewall enabled switch.

FIG. **2***a* is a schematic view of an L2 firewall enabled switch.

FIG. **2***b* shows an exemplary communication network including plural zones partitioned by a single security switch.

FIG. **3** is a flow diagram of a method for processing packets in the security switch of FIG. **2***a*.

FIG. **4** is a flow diagram for a method for processing un-recognized packets in the security switch of FIG. **2***a*.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Referring now to FIG. **1**, a packet switch communication network **100** includes a plurality of terminal units **102** configured in a plurality of zones **104** and coupled by one or more switches **106**.

In one implementation, each terminal unit **102** is of the form of a standalone computer (e.g., a personal computer, a laptop or workstation). Alternatively, one or more terminal units may be of the form of a personal digital assistant (PDA), Web pad, two-way pager, cellular handset, or other termination or remote device in a communication or computing environment. In one implementation, each terminal is a gateway to another network or group of terminal units (e.g., to a LAN or a pool of servers).

Each zone **104** embodies a security domain in the communication system. Each security domain can include separate policy, traffic management, accounting and administrative definitions and functions. Security policies, traffic management and other filtering functions can be enforced among and within zones. In one implementation, security policies are enforced between zones, while intra-zone communications are not subject to the security constraints. In one implementation, zones overlap. When zones overlap, policies associated with a parent zone can be a superset of the policies associated with one or more sub-zones (each including a subset of the overall policies). Alternatively, the policies associated with the parent zone may be separate and distinct from the policies of each sub-zone. For example, in one implementation, a zone can include one or more sub-zones, each including a separate set of policies.

In one implementation, each zone is associated with physical boundaries or other segmentation in the communication network. Alternatively, the assignment of particular terminal units to zones may represent groupings or combinations in a business structure (e.g., zones used to separate different functional entities in a business organization). Alternatively, the zones have no particular relation to physical boundaries. Communication between terminal units in each zone and among terminal units within a zone are controlled in accordance with protocols described below in association with switch **106**.

Switch **106** may be of different types. In one implementation, each switch **106** is configured as a layer **2** (L2) device and includes a plurality of ports on which packets from the communication network are received and transferred in accordance with L2 protocols. Each switch **106** includes a media access connection (MAC) table for use in determining switching of received packets. The MAC table associates MAC addresses with ports of the switch **106**. Packets are processed as they arrive at the ports of each switch **106** in accordance with L2 header information contained within a given packet. Depending on the MAC address, packets are switched to an appropriate output port as specified in the MAC table.

One or more of switches **106** are configured to enforce security domain constraints. For example, one or more of switches **106** is configured as an L2 firewall enabled security switch (hereinafter "security switch"). Referring now to FIG. **2**, a security switch **200** includes a plurality of ports **202**, a switch fabric **220** and an L2 controller **230**. Each port **202** is coupled to a security controller **204** by a bus **206**. The security controller **204** is coupled to one or more storage elements **208**. In one implementation (not shown), each port **202** is associated with a separate security controller **204** and storage element **208**. Alternatively, the security controller functionality can be combined in a single (as shown) or lesser number of individual security controller units. In addition, packets associated with all ports **202** can be stored in a single memory element **208** (as shown). Security switch **200** also includes a firewall device **210** that is coupled to (each) storage element **208** by a security bus **211**.

L2 controller **230** supports L2 switching protocols. Packets are either directly processed (e.g., intra-zone packets) or processed after a security screening (e.g., for inter-zone packets) as discussed in greater detail below. Associated with L2 controller **230** is a MAC table **235**. MAC table **235** includes plural entries each of which includes a MAC address and an indicator of a port **202** associated therewith. Switch fabric **220** is used to route traffic from storage element **208** to a respective port **202** under the control of L2 controller **230** using bus **221**.

Storage element **208** is partitioned into two portions. A first portion **215** is used to store packets received from a port **202** that are not subject to security screening. For example, in one implementation, packets received from a terminal unit in a same security zone (e.g., intra-zone traffic) are not subject to security screening. Un-screened packets are processed directly by L2 controller **230** and forwarded out a designated port in accordance with L2 protocols as specified in MAC table **235**. Second portion **217** is used to store packets to be screened by firewall device **210**.

Security controller **204** includes a screening engine **240**. Screening engine **240** examines each packet received from a respective port **202** and determines whether security screening is to be performed. In one implementation, screening engine **240** examines the L2 header for each packet, and based on the screening, either forwards the packet to the first or second portion **215** and **217**, respectively, of storage element **208**. The L2 header includes a destination MAC address that can be mapped to an egress port on the device using the MAC table **235**. Associated with each ingress and egress port is a security zone identifier. Security zone identifiers can be stored in a table of zone identifiers (not shown) that is indexed by port identifier (id). Screening engine **240** compares the security zone identifier associated with the packet being processed (determined from the identification of the egress port from the MAC table using the destination MAC address in the header of the packet being processed) with the security zone identifier associated with the port on which the packet was received in the device. Based on the comparison, screening engine **240** can determine whether the packet is destined for another zone (i.e., constitutes intra-zone or inter-zone communication).

The screening of packets can be with or without the knowledge of the individual terminal units. Associated with security switch **200** is a user interface (not shown) and associated management tools (not shown) for constructing one or more security zones. In one implementation, the security zones are determined based on the destination MAC address included in the L2 header of the packet received. More specifically, each egress port can be assigned to a security zone and have an associated security zone identifier associated therewith. Alternatively, the security zones can be created for plural users coupled to different ports of the security switch **200**. For example, security switch **200** can be configured to include three ports, where terminal units associated with a first two of the ports are assigned to a first zone, while terminal units associated with the third port are assigned to a second zone. Other configurations are possible. Zone assignments and partitions are discussed in greater detail below. The user interface allows an administrator or user to configure the security switch **200**. The security switch **200** can be configured to create plural security zones and associate one or more interfaces with each zone. Thereafter, policies can be established for inspecting or otherwise screening packets as they traverse the security switch **200**.

Firewall device **208** includes plural engines for performing packet screening prior to routing packets through security switch **200**. Firewall device **208** includes a firewall engine **270** and associated policies **271**, authentication engine **272**, encryption engine **274**, decryption engine **276** and a firewall controller **278**.

Firewall controller **278** extracts packets from second portion **217** of storage element **208**. Firewall controller **278** oversees the distribution of packets within the firewall device as well as the coordination among the respective engines. Each packet is evaluated and processed in accordance with policies based on one or more considerations. For example, packets can be screened based on source, destination or both. One or more policies **271** are retrieved and used by firewall engine **270** to inspect the packet. Packet inspection may also require encryption, decryption and authentication services. One or more of the encryption **272**, decryption **274** and authentication **276** engines can be invoked by the firewall controller **278** as part of the inspection processes. In addition, other services can be provided including virtual private network termination services, session set-up and various other traffic management and security related functions. Examples of screening services are discussed in greater detail below. After the inspection, packets can be forwarded in the network or dropped as appropriate. In one implementation, packets that are to be forwarded (e.g., pass the inspection) are prepared as appropriate (e.g., encrypted) then forwarded to the first portion **215** of storage element **208**. Alternatively, the packets may be returned to the second portion **217** of storage element **208** and marked as having been screened. In one implementation, screened packets are forwarded to a queue for processing by L2 controller **230**. Screened packets are then processed by L2 controller **230** and switched to an appropriate output port in accordance with conventional L2 processing protocols.

Each security switch **200** can be configured to create plural security zones. For example, a communications network having a security switch **200** is shown in FIG. **2***b*. The communications network is a VLAN structure that includes 3 zones. Security switch **200** includes a user interface and administrative control mechanisms for creating each of the security zones, specifying policies and other criteria for defining and managing each zone. The security zones enforced by the security switch **200** can be transparent to the end users. That is, the security zones can be established at the security switch **200** including the specification of all operating parameters associated with the security domain. Users in each zone may be unaware of the zone structure and may communicate with other users in a conventional manner. For example, a virtual private network can be created between users including encryption and decryption services without requiring the actual encryption and decryption support in the respective end users (e.g., encryption and decryption services can be provided in secure switches disposed between the two users). Accordingly, a system administrator can create a virtual private network between a remote user in one security zone and another user in a second security zone where the individual users are unaware of the VPN services and are not required to include encryption or decryption services locally. In one implementation, the administrator provisioned VPN services are specified for remote users in a same zone.

Alternatively, the users may be aware of the security structure and include indicators (e.g., zone identifiers) in packets transferred to other users. Each user may define their own custom L2 zone and an inter-zone policy for their network security requirements. For example, security switch **200** shown in FIG. **2***b* embodies a VLAN that includes v**1**-trust, v**1**-untrust and v**1**-dmz zones. V**1**-trust defines a zone that includes two users including user **291** and user **292**. V**1**-untrust defines a zone that includes a single user **293**. V**1**-dmz defines a zone that includes three users, users **291**, **292** and user **294**. Separate policies can be enforced for communications between the three zones. For example, communications that are intra-zone between user **291** and user **292** will not require inspection, and as such are handled by security switch **200** in accordance with conventional L2 protocols. Communications from user **291** to user **293** will invoke an inspection process as defined by the security system architect (e.g., user **291** or **292** or an administrator for such) for communications between V**1**-trust and V**1**-untrust. Similarly, communications between user **294** and user **291** will invoke an inspection process (e.g., a potentially lesser screen) for communications between V**1**-dmz and V**1**-trust.

Multiple interfaces are allowed inside each zone. For intra-zone traffic, security switch **200** behaves like a tradition L2 bridge forwarding a given packet based on the destination MAC-address. In one implementation, no firewall protection mechanisms are applied for the intra-zone traffic.

For inter-zone traffic, standard firewall inspections (including policy inspection, TCP stateful inspection, etc. as described above) are performed for each incoming packet. In all cases, the egress interface is determined by the learned destination MAC address on the interface.

Packet Flow

Referring now to FIG. **3**, a method **300** is shown, as invoked by the security switch **200**, for processing packets. The method described is made with no particular reference to the specific hardware elements performing the steps. An exemplary hardware configuration is given above. The method can however be implemented in L2 switches having other configurations. The method begins with the receipt of a packet (**302**). The packet is evaluated to determine whether the packet is to be inspected (**304**). If so, the packet is pre-processed as appropriate (**305**) and one or more policies are retrieved (**306**). The pre-processing of the packet can include decryption and authentication services. The retrieval of a policy includes the identification of the zone to which the packet is being transferred. Packets traveling between zones can be inspected using a security policy. Intra-zone communications may not be inspected. In one implementation, policies can be enforced on intra-zone communications. The retrieval of a policy includes a MAC look-up for the MAC destination address in a received packet in the MAC table to determine an egress port associated with the MAC address and necessarily a security zone. The security zones associated with the packet's ingress and egress ports are compared to determine if the packet is passing to another zone. Assuming that an inspection is to occur, an appropriate policy is retrieved (i.e., based on the ingress port and egress port identifiers and their respective security zones). Thereafter, the packet is inspected (**308**). Packet inspection can include screening and dropping the packet as required. If the packet is to be forwarded on the network (**309**), post-processing operations are invoked as appropriate (**310**). Alternatively, the packet is dropped (**311**). The post processing operations can include session set-up, encryption and other functions. Thereafter the packet is processed in accordance with conventional L2 protocols starting at step **312**.

At step **312**, either a packet has passed inspection or did not require inspection. In either case, L2 header information is extracted to determine a MAC address associated with the packet. A look-up of the MAC address is performed (**314**) and the packet is then routed to an appropriate output port (**316**). Thereafter the process ends.

Referring again to FIG. **2**, the process steps are described with reference to one hardware implementation of the invention. Packets are received at a port **202**. Each packet is transferred on bus **205** to, and routed through, security controller **204** and stored in storage element **208** via a storage bus **209**. Security controller **204** evaluates each packet to determine if inspection is required and forwards the packets to an appropriate portion of storage device **208**. Packets that are not to be inspected (i.e., packets stored in first portion **215** of storage device **208**) are processed by L2 controller **230**. When L2 controller **230** is available, packets are fetched and processed to determine a port to which the packet should be forwarded. L2 controller **230** evaluates the MAC address associated with the packet, and using MAC table **235**, determines a port for routing. After processing by the L2 controller **230**, the packet is forwarded to an appropriate link into switch fabric **220** for routing to a determined output port **202**.

Packets that are to be inspected are transferred by security controller **204** into second portion **217** of storage element **208**. When firewall engine **230** is available, a packet is fetched and processed to determine a security policy to be used in inspecting the packet. Firewall engine **270** evaluates IP address(es) associated with the packet and implements traffic control and management functions as appropriate. Packets that are to be forwarded (i.e., pass inspection) are returned to storage element **208**. Thereafter, the packet can be forwarded to an appropriate link into switch fabric **220** for routing to a determined output port **202**. Other packets are dropped or otherwise handled in accordance with the policies defined for the given security zones.

As discussed above, all packets that pass the inspection in the firewall device **210** as well as all packets that are not required to be inspected, are processed by L2 controller **230** in accordance with conventional L2 protocols. In one implementation, the processing of packets by L2 controller is modified to maintain security zones. More specifically, as discussed above, conventional L2 switches broadcast on all ports a packet that has a MAC address that is not recognized. This type of broadcast may well violate one or more security policies in place for given zones in the communication network. Accordingly, in one implementation a test packet is broadcast to each port. The broadcasting of test packets is described in more detail in association with FIG. **4**.

Referring now to FIG. **4**, a method **400** is shown for handling packets by the L2 controller and includes receiving a packet to be processed (**402**). The MAC address for the packet is extracted (**404**). A check is made to locate an entry in a MAC address table that corresponds to the extracted MAC address (**406**). If a match is located (**407**), the packet is routed to an output port associated with the matching entry (**408**). If no match is located, the packet is dropped (**410**). In one implementation, the packet is merely held for a predetermined amount of time in hope of receiving information regarding the non-matching MAC address. If no match is located, a probe packet is created (**412**). The probe packet includes the MAC address associated with the packet being processed (i.e., the original ingress packet). In one implementation, the probe packet is an "ICMP PING" packet with an IP TTL field set to 1. Each packet includes the same MAC addresses (L2) and source/destination IPs (L3) as the ingress packet whose MAC address could not be located. The probe packet is then broadcast to all ports (**414**). A check is made to determine if a response is received on any of the security device's ports (**416**). The ICMP PING packet will cause the right gateway, which was to receive and forward the original ingress packet, to respond to the L2 controller in the device with an "ICMP TTL expired" message packet. From the expired packet, the system can identify the proper egress port/zone associated with the received MAC address. This method guarantees that no information in the original ingress packet will be leaked out. If a response is received (indicating that a device coupled to the receiving port is configured to process packets having the identified MAC address), then the MAC table is updated to include an entry having the MAC address and a port identifier indicating the port on which the response was received (**418**). Thereafter the process ends.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the firewall device has been described in terms of screening at the L3 layer level. Alternatively, other screening can be invoked at other levels including layers up to and including layer **7** (L7) processing. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. An L2 device comprising:

at least one port to couple to a terminal unit included in a first security zone;

at least one port to couple to a terminal unit included in a second security zone that is distinct from the first security zone;

a controller to determine for each packet received from either the first security zone or the second security zone whether the received packet is an inter-zone packet destined for the other of the first security zone or the second security zone;

a firewall engine to inspect and filter received inter-zone packets using a zone specific policy; and

an L2 switching engine to transfer to a port associated with intra-zone transfer, without inspection by the firewall engine, received intra-zone packets using a table of MAC addresses and corresponding ports, and to transfer to a port associated with inter-zone transfer, inter-zone packets that are retained after the inspection by the firewall engine.

2. An L2 device comprising:

a controller to determine for each packet received whether the received packet is to be transferred intra-zone or inter-zone, each zone representing a distinct security domain and having an associated policy for use in inspecting packets entering and exiting an associated zone;

a firewall engine to inspect and filter received inter-zone packets using a zone specific policy; and

an L2 switching engine operable to:

route to an intra-zone port, without the inspection by the firewall engine, received intra-zone packets using a table of MAC addresses and corresponding ports, and

route to an inter-zone port inspected inter-zone packets that are retained after the inspection by the firewall engine.

3. An L2 device comprising:

a controller to determine for each packet received whether the received packet is to be transferred inter-zone or intra-zone, inter-zone being between a plurality of zones and intra-zone being between a single one of the zones, each zone representing a distinct security domain; and

a firewall engine to inspect and filter inter-zone packets using a zone specific policy prior to permitting inter-zone routing using L2 protocols, wherein intra-zone packets are not inspected by the firewall engine.

4. An L2 device comprising:

a controller to determine for each packet received whether the received packet is an inter-zone packet that is permitted to be transferred from a first distinct security domain to a second distinct security domain subject to a security inspection or an intra-zone packet that is permitted to be transferred within the first or second distinct security domain without being subjected to a security inspection; and

an inspection device to inspect and filter inter-zone packets using a zone specific policy prior to inter-zone routing using L2 protocols.

5. An L2 device comprising:

a controller to determine for each packet received whether the received packet is to be inspected against a security policy;

an inspection device to inspect and filter only those packets identified by the controller as needing inspection based on a zone specific policy; and

an L2 controller to transfer inspected packets from a first security zone to a second security zone in accordance with L2 header information using L2 protocols, and transfer non-inspected packets within the first or second security zones.

6. The device of claim **5** wherein the inspection device is a firewall.

7. The device of claim **5** wherein the inspection device is a layer **3** firewall device.

8. The device of claim **5** wherein the inspection device is a layer **4** firewall device.

9. The device of claim **5** wherein the inspection device is a layer **7** firewall device.

10. The device of claim **5** wherein the inspection device is a firewall that filters based on layer information other than layer **2** header information.

11. The device of claim **5** wherein the controller determines each packet that is to pass between security zones and the inspection device only processes inter-zone traffic.

12. The device of claim **5** wherein the controller determines each packet that is to remain in a single security zone and transfers intra-zone packets to the L2 controller, bypassing the inspection device.

13. The device of claim **12** wherein the device uses a MAC address in the layer **2** header of a given packet to determine an egress port on the device to which the packet is to be transferred.

14. The device of claim **5** further comprising a storage element for storing packets that are to be inspected and an L2 controller transferring packets through the device including determining an egress port for transferring a given packet using a destination MAC address in the given packet and a MAC address table that includes a mapping of MAC addresses and associated egress nodes.

15. The device of claim **14** wherein the memory element includes a first and second portion, the first portion storing packets to be transferred through the device, and the second portion storing packets waiting for inspection.

16. The device of claim **5** wherein the device is an L2 switch.

17. The device of claim **5** wherein the device is an L2 bridge.

18. A method for transferring packets in a communication network, the method comprising:

receiving a packet at an L2 device;

determining whether the received packet is an intra-zone packet to be transferred within a single zone or an inter-zone packet to be transferred between zones, each zone representing a distinct security domain;

inspecting and filtering inter-zone packets using a zone specific policy prior to inter-zone routing of the inter-zone packets using L2 protocols; and

routing the ultra-zone packets without being subject to security inspection or filtering.

19. A method for transferring packets in a communication network, the method comprising:

receiving a packet at an L2 device;

determining whether the received packet is to be inspected against a security policy;

inspecting and filtering identified packets using a zone specific policy prior to transferring the packet from a first security zone through the L2 device using L2 protocols to a second security zone distinct from the first security zone; and

transferring non-inspected packets either from the first security zone to the first security zone, or from the second security zone to the second security zone.

20. A method for switching packets in a communication network including plural zones, each zone representing a distinct security domain, the method comprising:

receiving a packet at an interface of an L2 device;

determining if a destination MAC address associated with the received packet is known; and

if not,

holding the received packet a predetermined amount of time without transferring the packet to any port of the L2 device,

creating a probe packet that includes the unknown MAC address, and

broadcasting the probe packet to all interfaces except the receiving interface.

21. The method of claim **20** wherein the probe packet includes a time to life (TTL) field in an IP header and the method includes setting a value of the TTL field such that a downstream node having the unknown MAC address and receiving the probe packet will return an expired message to the L2 device.

22. The method of claim **20** further comprising dropping the packet after the expiration of the predetermined amount of time.

23. The method of claim **20** wherein the packet is dropped if the MAC address is unknown.

24. The method of claim **20** further comprising receiving a response from one of the broadcast interfaces and updating a table indicating a previously unknown MAC address is associated with the responding interface.

* * * * *

EXHIBIT E

US006772347B1

(12) **United States Patent**
Xie et al.

(10) **Patent No.:** **US 6,772,347 B1**
(45) **Date of Patent:** **Aug. 3, 2004**

(54) **METHOD, APPARATUS AND COMPUTER PROGRAM PRODUCT FOR A NETWORK FIREWALL**

(75) Inventors: **Ken Xie**, Atherton, CA (US); **Yan Ke**, San Jose, CA (US); **Yuming Mao**, Milpitas, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/525,369**

(22) Filed: **Mar. 15, 2000**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/283,730, filed on Apr. 1, 1999, now Pat. No. 6,701,432.

(51) **Int. Cl.**^{7} **G06F 13/00**; G06F 15/16

(52) **U.S. Cl.** **713/201**; 713/154; 709/242; 709/243

(58) **Field of Search** 713/200, 201, 713/154, 155, 160; 709/243, 242, 238

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,606,668 A		2/1997	Shwed	395/200.11
5,835,726 A		11/1998	Shwed et al.	395/200.59
5,951,651 A	*	9/1999	Lakshman et al.	709/239
6,009,475 A	*	12/1999	Shrader	709/249
6,016,310 A	*	1/2000	Muller et al.	370/255
6,400,707 B1	*	6/2002	Baum et al.	370/352
2002/0188720 A1	*	12/2002	Terrel et al.	709/225

FOREIGN PATENT DOCUMENTS

EP	000658837 A2	*	6/1995	G06F/1/00
EP	000893921a1	*	4/1999	H04N/7/67

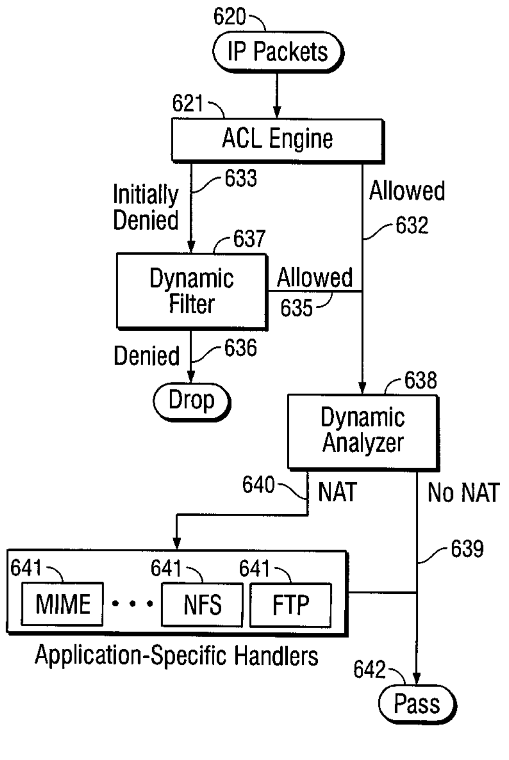
* cited by examiner

Primary Examiner—Norman M. Wright
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

Systems and methods for network security including a firewall. One firewall includes a firewall engine. The firewall engine includes a first engine including a first set of rules for sorting incoming IP packets into initially allowed packets and initially denied packets. The firewall engine also includes a filter including a second set of rules for receiving and further sorting the initially denied packets into allowed packets and denied packets.

24 Claims, 6 Drawing Sheets



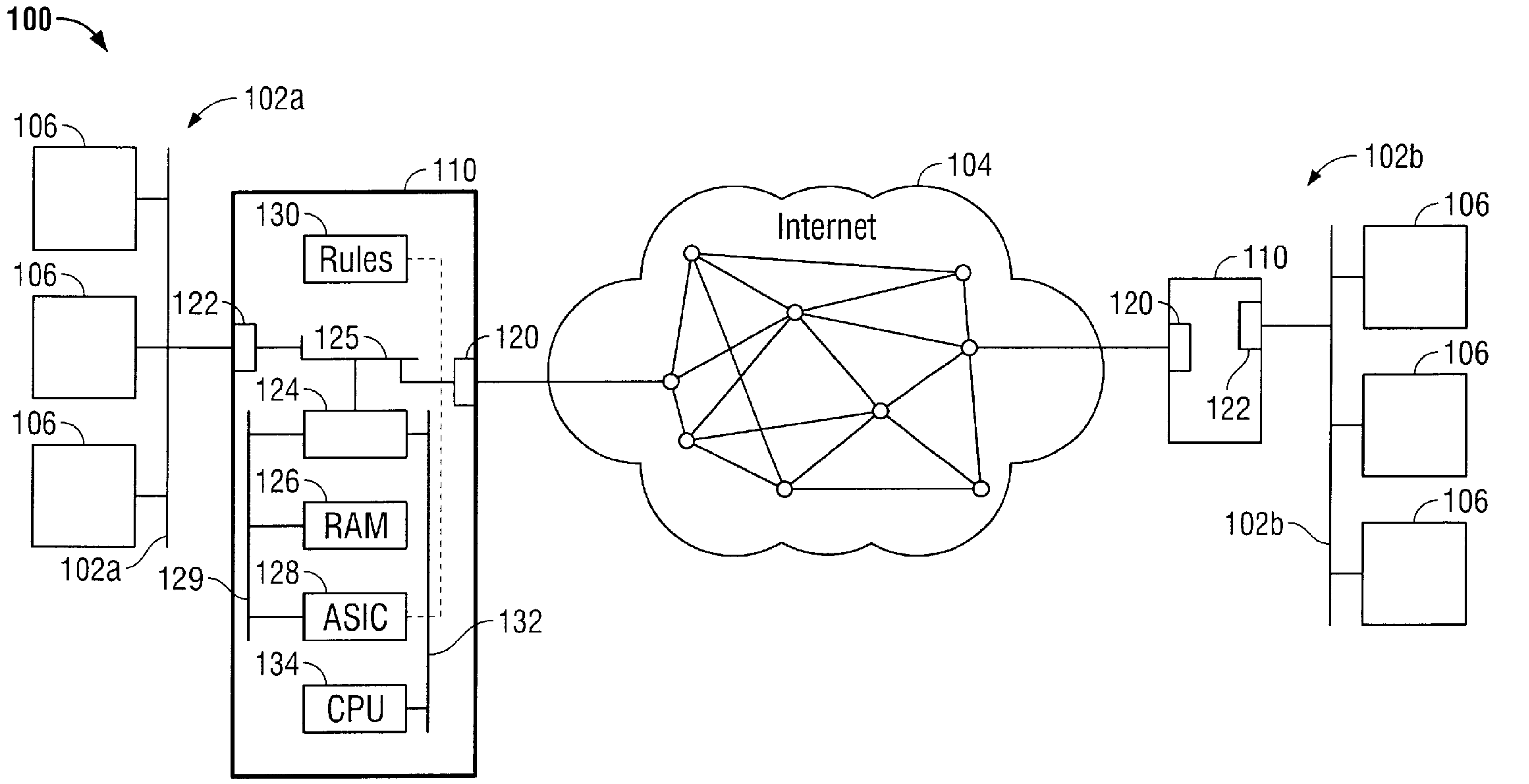

100
102a
106
106
106
102a
110
130
Rules
122
125
120
124
126
RAM
129
128
ASIC
132
134
CPU
104
Internet
102b
110
120
122
106
106
106
102b

FIG. 1

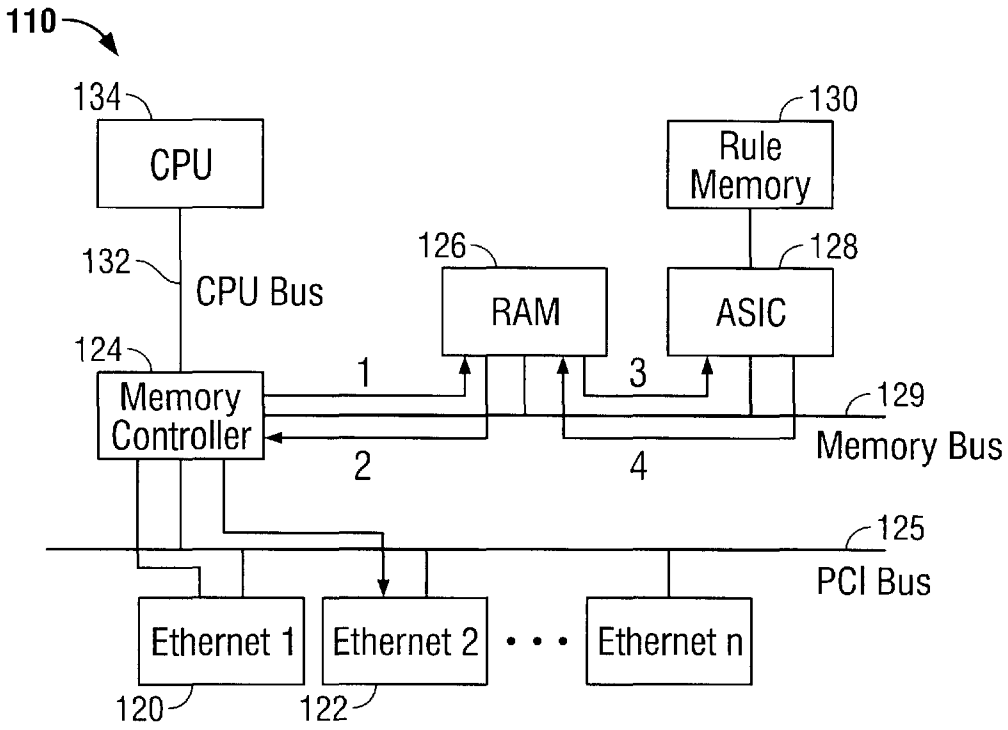

110
134
CPU
130
Rule Memory
132
CPU Bus
126
RAM
128
ASIC
124
Memory Controller
1
3
129
2
4
Memory Bus
125
PCI Bus
Ethernet 1
Ethernet 2
Ethernet n
120
122

FIG. 2A

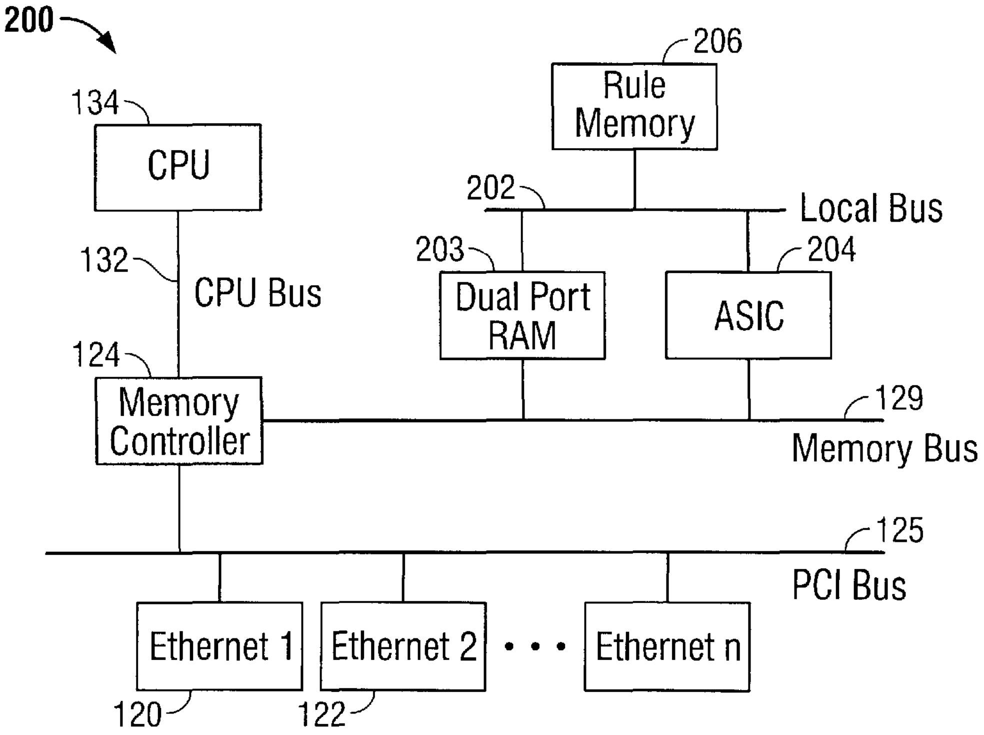

200
206
Rule Memory
134
CPU
202
Local Bus
203
204
132
CPU Bus
Dual Port RAM
ASIC
124
Memory Controller
129
Memory Bus
125
PCI Bus
Ethernet 1
Ethernet 2
Ethernet n
120
122

FIG. 2B

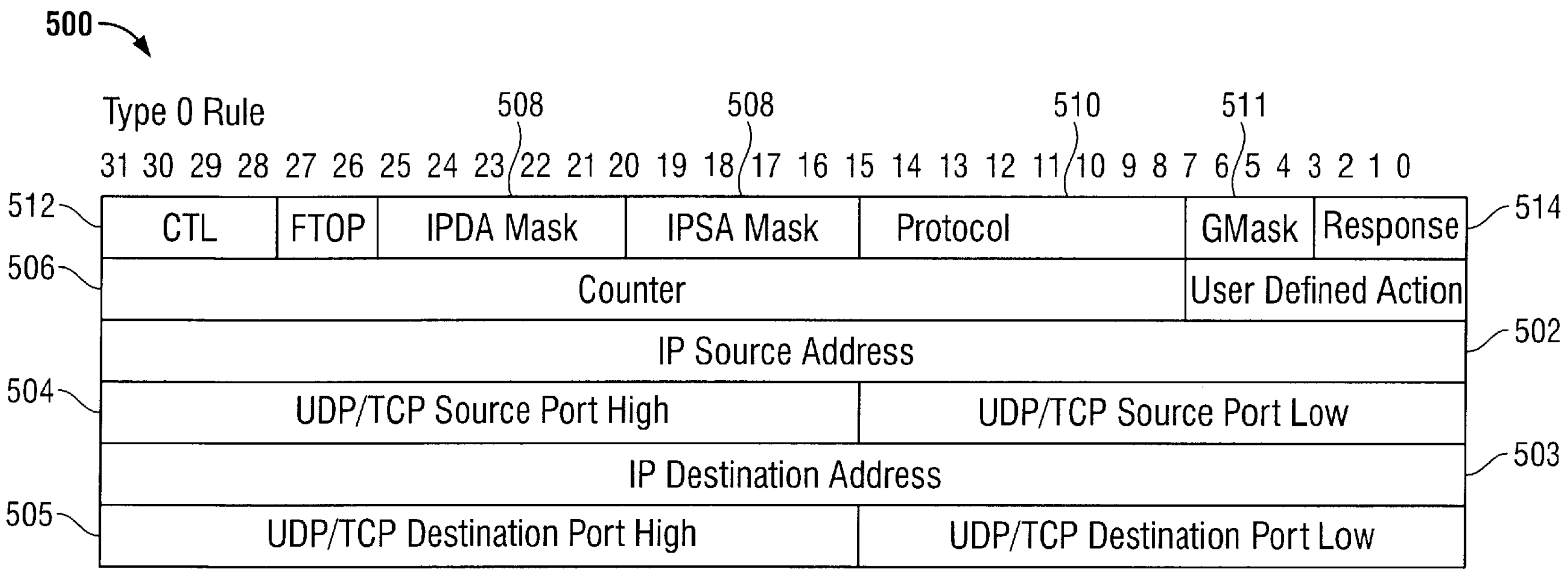

500
Type 0 Rule
508
508
510
511
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
512
CTL
FTOP
IPDA Mask
IPSA Mask
Protocol
GMask
Response
514
506
Counter
User Defined Action
IP Source Address
502
504
UDP/TCP Source Port High
UDP/TCP Source Port Low
IP Destination Address
503
505
UDP/TCP Destination Port High
UDP/TCP Destination Port Low

FIG. 3

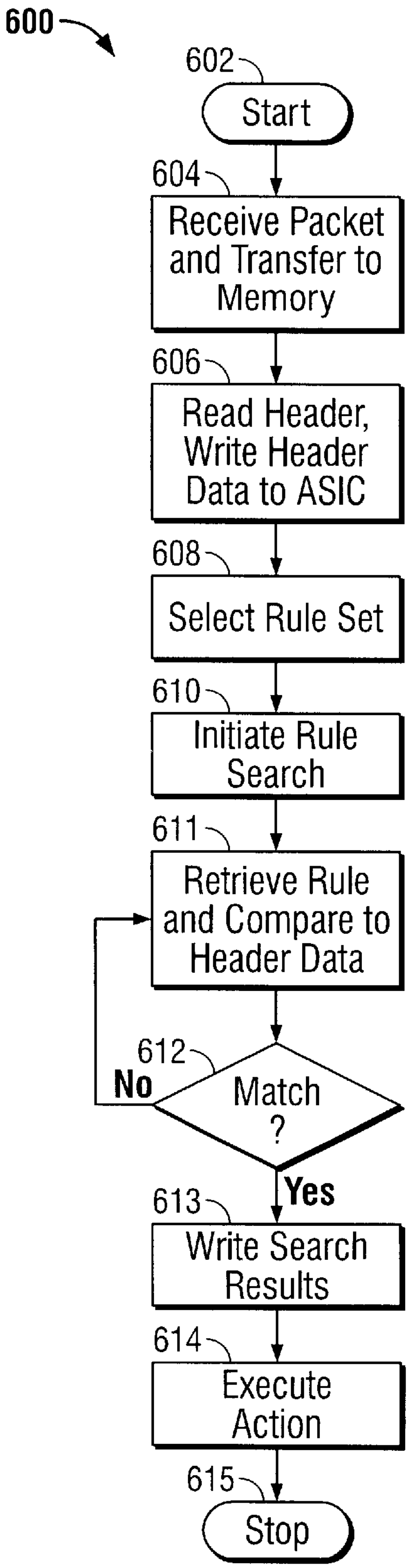

600
602
Start
604
Receive Packet and Transfer to Memory
606
Read Header, Write Header Data to ASIC
608
Select Rule Set
610
Initiate Rule Search
611
Retrieve Rule and Compare to Header Data
612
Match ?
No
Yes
613
Write Search Results
614
Execute Action
615
Stop

FIG. 4

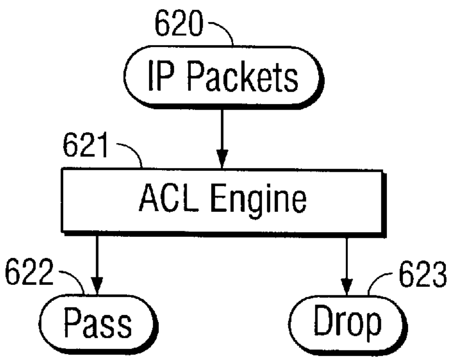


FIG. 5
(Prior Art)

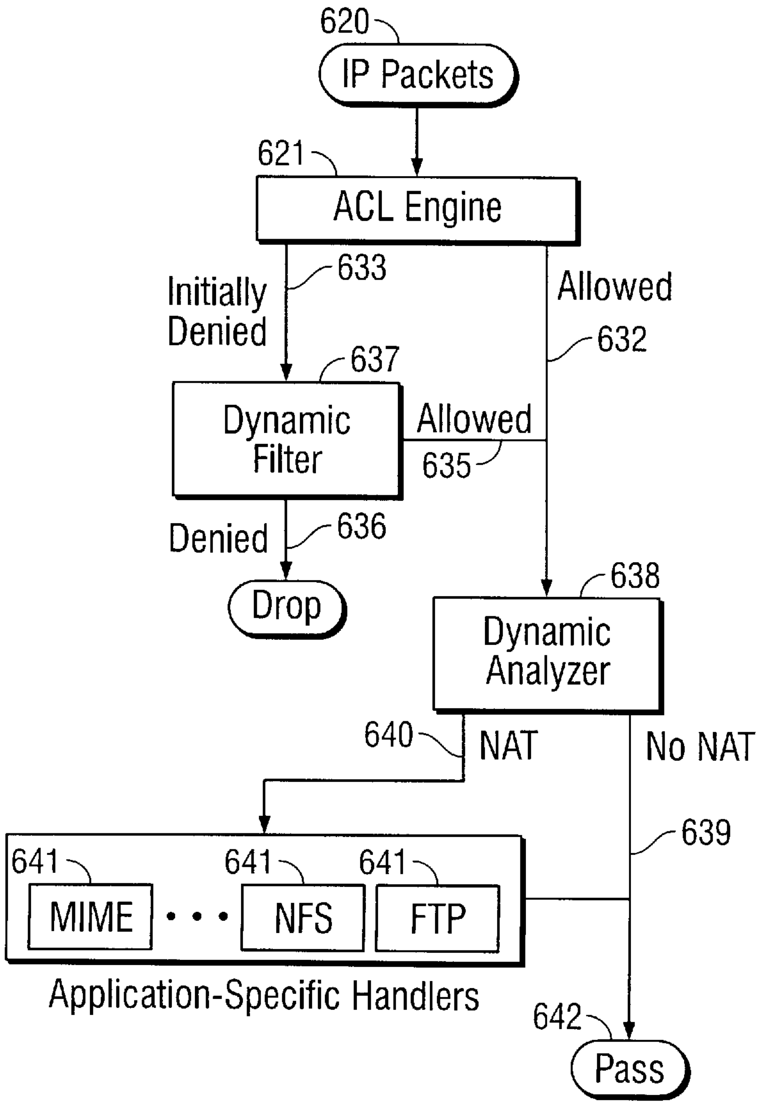


FIG. 6

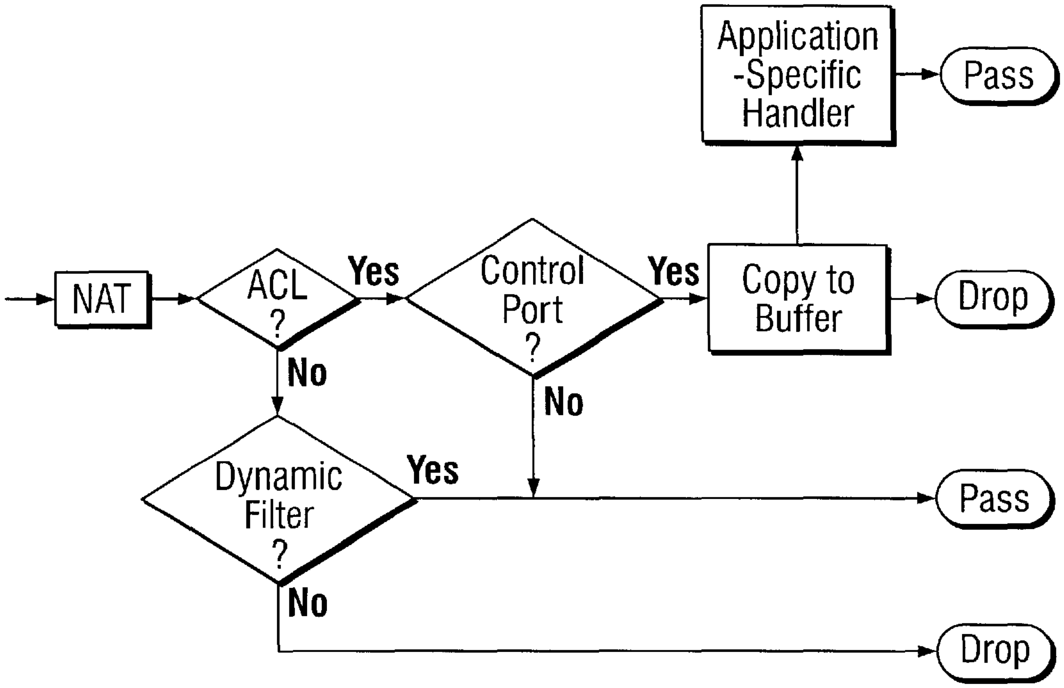

NAT
ACL ?
Yes
Control Port ?
Yes
Copy to Buffer
Application -Specific Handler
Pass
Drop
No
No
Dynamic Filter ?
Yes
Pass
No
Drop

FIG. 7A

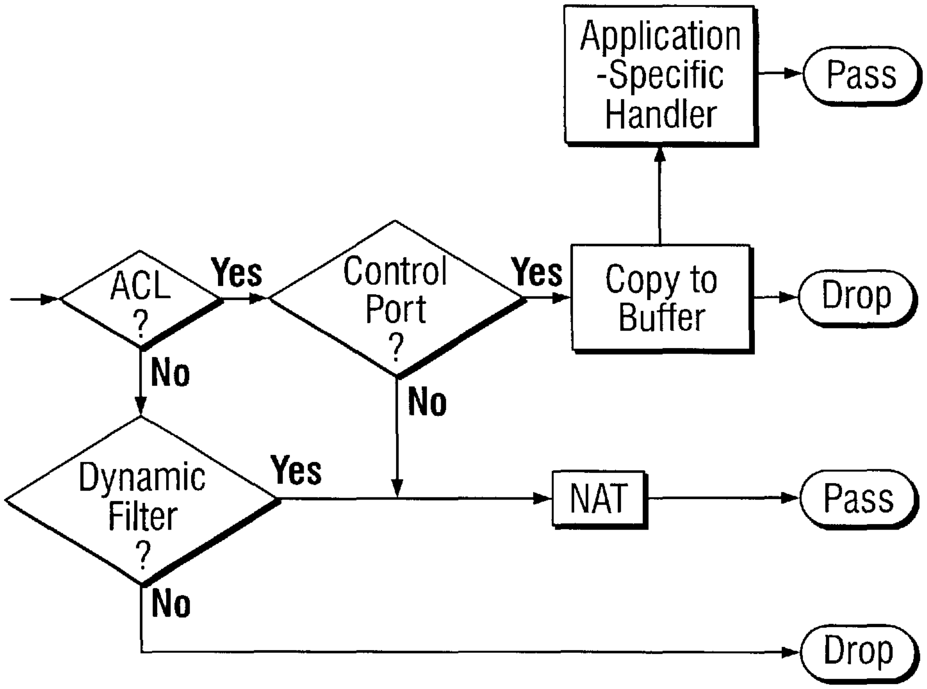

ACL ?
Yes
Control Port ?
Yes
Copy to Buffer
Application -Specific Handler
Pass
Drop
No
No
Dynamic Filter ?
Yes
NAT
Pass
No
Drop

FIG. 7B

METHOD, APPARATUS AND COMPUTER PROGRAM PRODUCT FOR A NETWORK FIREWALL

RELATED APPLICATIONS

The present application is a continuation-in-part of application Ser. No. 09/283,730 now U.S. Pat. No. 6,701,432 filed Apr. 1, 1999.

FIELD OF THE INVENTION

The present invention relates to the field of computer networks. In particular, the present invention relates to a method, apparatus and computer program product for providing network security.

BACKGROUND OF THE INVENTION

A packet switch communication system includes a network of one or more routers connecting a plurality of users. A packet is the fundamental unit of transfer in the packet switch communication system. A user can be an individual user terminal or another network. A router is a switching device that receives packets containing data or control information on one port and, based on destination information contained within the packets, routes the packets out another port to their final destination, or to some intermediary destination(s). Conventional routers perform this switching function by evaluating header information contained within the packet in order to determine the proper output port for a particular packet.

As known, a communications network can be a public network, such as the Internet, in which data packets are passed between users over untrusted, i.e., nonsecure communication links. Alternatively, various organizations, typically corporations, use what is known as an intranet communications network, accessible only by the organization's members, employees, or others having access authorization. Intranets typically connect one or more private servers, such as a local area network (LAN). The network configuration in a preferred embodiment of this invention can include a combination of public and private networks. For example, two or more LANs can be coupled together with individual terminals using a public network, such as the Internet. A network point that acts as an entrance to another network is known in the art as a gateway.

Conventional packet switched communication systems that include links between public and private networks typically include means to safeguard the private networks against intrusions through the gateway provided at the interface of the private and public networks. The means designed to prevent unauthorized access to or from a private are commonly known as firewalls, which can be implemented in both hardware and software, or a combination of both. Thus, a firewall is a device that can be coupled in-line between a public network and a private network for screening packets received from the public network.

Referring to FIG. **1**, a conventional packet switch communication system **100** can include two (or more) private networks **102***a* and **102***b* coupled by a public network **104** for facilitating the communication between a plurality of user terminals **106**. Each private network **102** can include one or more servers and a plurality of individual terminals. Each private network **102** can be an intranet, such as a LAN. Public network **104** can be the Internet, or other public network having untrusted links for linking packets between private networks **102***a* and **102***b*. In a preferred embodiment, at each gateway between a private network **102** and public network **104** there is a firewall **110**.

The architecture of an illustrative prior art firewall is shown in FIG. **2***a*. The firewall **110** generally includes one

or more public network links **120**, one or more private network links **122**, and memory controller **124** coupled to the network links by a PCI bus **125**. Memory controller **124** is also coupled by a memory bus **129** to a memory (RAM) **126** and a firewall engine, implemented in a preferred embodiment as an ASIC **128**. The firewall engine ASIC **128** performs packet screening prior to routing packets through to private network **102**. The firewall engine ASIC **128** processes the packets to enforce an access control policy, screening the packets in accordance with one or more sets of rules. The rules are described in more detail below. A central processor (CPU) **134** is coupled to memory controller **124** by a CPU bus **132**. CPU **134** oversees the memory transfer operations on all buses shown. Memory controller **124** is a bridge connecting CPU bus **132**, memory bus **129**, and PCI bus **125**.

In operation, packets are received at public network link **120**. Each packet is transferred on bus **125** to, and routed through, memory controller **124** and on to RAM **126** via memory bus **129**. When firewall engine **128** is available, packets are fetched. using memory bus **129** and processed by the firewall engine **128**. After processing, the packet is returned to RAM **126** using memory bus **129**. Finally the packet is retrieved by the memory controller **124** using memory bus **129**, and routed to private network link **122**. The screening rules implemented by the firewall engine **128** are typically searched in linear order, beginning with the internal rule memory. Certain aspects of the rule structure are described below.

As known in the art, a rule is a control policy for filtering incoming and outgoing packets. Rules specify actions to be applied as against certain packets. When a packet is received for processing through a rule search, the packet's IP header, TCP header, or UDP header may require inspecting. A rule will generally include, at a minimum, source/destination IP addresses, UDP/TCP source/destination ports and transport layer protocol. Additional criteria may be used by the rules as well.

Generally, the address information is used as matching criterion—in other words to match a rule, a packet must have come from a defined source IP address and its destination must be the defined destination IP address. The UDP/TCP source/destination port specifies what client or server process the packet originates from on the source machine. The firewall engine can be configured to permit or deny a packet based upon these port numbers. The rule may include a range of values or a specific value for a TCP/UDP port. The transport layer protocol specifies which protocol above the IP layer, such as TCP or UDP, the policy rule is to be enforced against.

The firewall engine described above essentially screens packets using an access control list (ACL), and may be referred to as an ACL engine. That is, it performs a simple comparison of various matching criteria of an incoming IP packet—typically source, destination, port and protocol—to each rule in a rule set in sequence. Based upon this comparison, an incoming IP packet is either allowed or denied. A data-flow chart for this firewall engine is shown in FIG. **5**.

It will be appreciated that using a fixed set of rules can be restrictive in many practical applications. Therefore, it is desirable to provide a system and method capable of adding rules to the rule set of the firewall engine dynamically—that is, to extract from a sequence of packets information, such as the port number and IP address, and generate new rules using this information. However, generating these new rules dynamically would increase the complexity of the comparison and decrease the speed of the firewall engine. There is therefore a need in the art for a firewall engine which can generate rules dynamically, based upon information extracted from incoming packets, with a limited impact on the speed of the firewall engine.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment, an apparatus, method and computer program product for providing network security is described. The apparatus includes an engine for sorting incoming IP packets into initially allowed and initially denied packets using a fixed set of rules. The packets are then further sorted by a second engine. In one embodiment, the engine further sorts the initially denied packets into allowed packets and denied packets, using dynamically generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

Likewise, the method includes the step of sorting incoming IP packets into initially allowed and initially denied packets using a fixed set of rules. The packets are then further sorted. In one embodiment, additional steps include sorting the initially denied packets into allowed packets and denied packets, using dynamically generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

Finally, the computer program product sorts incoming IP packets into initially allowed packets and initially denied packets. In one embodiment, the computer program product further sorts the initially denied packets into allowed packets and denied packets, using dynamically generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates an exemplary packet switch communications system.

FIG. **2***a* illustrates a firewall with an application-specific integrated circuit (ASIC).

FIG. **2***b* illustrates a firewall with a local bus and an application-specific integrated circuit (ASIC).

FIG. **3** illustrates an exemplary rule structure for use by a firewall.

FIG. **4** is a flow diagram for a firewall screening process.

FIG. **5** is a data-flow chart for a prior art firewall.

FIG. **6** is a data-flow chart for a firewall in accordance with one embodiment of the invention.

FIG. **7***a* is a logic diagram for processing incoming packets in accordance with the invention.

FIG. **7***b* is a logic diagram for processing outgoing packets in accordance with the invention.

DETAILED DESCRIPTION OF THE INVENTION

A conventional firewall may be implemented in software, or in hardware as shown in FIG. **2***a*. Alternatively, a hybrid of software and hardware may also be used to implement a firewall. The firewall of FIG. **2***a* uses a memory bus **129** to communicate between the ASIC **128**, the RAM **126**, and the memory **130**, which stores the rules used by the firewall. FIG. **2***b* shows a high-speed firewall that employs a local bus **202** for improved processing speed. A high-speed firewall is described in pending parent application Ser. No. 09/283,730, the contents of which is hereby incorporated by reference. Exemplary high-speed firewalls include NetScreen Technology, Inc.'s integrated firewall products, described at www.netscreen.com and related web pages. Selected web pages describing NetScreen's high-speed firewalls are provided as Appendix A to this application.

As shown in FIG. **2***b*, the high-speed firewall includes-a hardware ASIC **204** to implement the firewall engine. The firewall engine retrieves packets stored in memory and processes each packet to enforce an access control policy. The processing by the firewall engine includes retrieving rules from a rule set, and screening the packets in accordance with the retrieved rules. In a specific embodiment, the rules may be stored in an internal memory in the ASIC **204**, or may be retrieved from a separate rule memory **206** via the local bus **202**. In a preferred embodiment, frequently accessed rule sets may be stored in the internal memory, with less-frequently accessed rule sets being stored in the separate rule memory **206**.

The structure **500** of a rule used by a firewall engine in accordance with one embodiment of the present invention is shown in FIG. **3**. A rule will generally include, at a minimum, source/destination IP addresses **502 503**, UDP/TCP source/destination ports **504 505** and transport layer protocol **510**. Additional information used by the rules may include: a range of values for the UDP/TCP source/destination port **504 505**; a counter **506** to keep track of the number of times the rule has been matched; a general mask (GMASK) **511** to indicate whether to ignore or check certain information in the packet header; source/destination IP address mask **508** to indicate whether to ignore part of an IP address, typically a specified number of the least significant bits; a searching control field **512** to tell the firewall engine to search in the separate rule memory **206** and to give a starting address; and a response action field **514** to specify the action to be taken if the rule is matched.

The address information is used as matching criterion—to match a rule, a packet must have come from the defined source IP address **502** and its destination must be the defined destination IP address **503**. Part of the address may be masked using the source/destination IP address mask **508**. The UDP/TCP source/destination port **504 505** specifies what client or server process the packet originates from on the source machine. The firewall engine can be configured to permit or deny a packet based upon these port numbers. The rule may include a range of values or a specific value for a TCP/UDP port. The counter **506** is used to track the number of times a rule has been matched, and at some threshold value will trigger a certain action, such as deny, log or alarm. The transport layer protocol **510** specifies which protocol above the IP layer, such as TCP or UDP, the policy rule is to be enforced against.

Referring to FIGS. **2***b* and **4**, a process **600** executed by the firewall engine in the ASIC **204** is shown for screening packets using both the on-chip and off-chip rule memories. The firewall engine process begins at step **602**. A packet is received at an interface (public network interface **122**) and transferred to dual-ported memory **203** using a DMA process executed by memory controller **124** (**604**).

CPU **134** reads the packet header information from packet memory and writes the packet information into special registers on ASIC **204** (**606**). These registers are mapped onto the system memory space, so CPU **134** has direct access to them. In an exemplary hardware firewall, the registers include: a source IP register; a destination IP register; a port register; a protocol register; and an acknowledge register, for storing the acknowledge bit from the packet.

CPU **134** also specifies which rule set to search by writing to a rule set specifier register (**608**). CPU **134** issues a command to the firewall engine located in the ASIC **204** by writing to a control register to initiate the ASIC rule search (**610**). Alternatively, the firewall engine may first check a stored look-up table with criteria relating to ongoing current applications or services, before searching the rules. In that case, the firewall engine first compares the contents of the special registers to the contents of a look-up table, where the look-up table includes the IP address, port and protocol corresponding to each current application or service. For example, if the packet is an FTP packet for an FTP that is ongoing, this information will be in the lookup table. If, on the other hand, the packet is an FTP packet for a newly-initiated FTP, the information will not be in the look-up table.

If the information is not in the look-up table, or if a look-up table is not used, the firewall engine then compares the contents of the special registers to each rule in sequence (**611**) until a match is found (**612**). The search stops when a match is found (**612**). Alternatively, for certain rules, known as counter rules, the firewall engine will increment the count register and continue the search. If the count threshold is exceeded, or if the search locates a match for a non-counter rule, the search results are written to a status register (**613**). Likewise, if no match is found, and the entire set of rules has been examined, the search results are written to the status register. In addition, when a match is found, if a look-up table is used the information identifying the current application, such as the IP address, port and protocol, are written to the look-up table so that later packets in the current application may be processed using the look-up table instead of a rule search.

During the search, CPU **134** polls the status register to check whether the firewall engine is busy or has completed the search. When the CPU **134** determines that the search is complete, the CPU **134** executes certain actions against the current packet based on the information in the status register, such as permit or deny the packet, signal an alarm, and log the packet (**614**).

The process described above is a prior art one-pass search process, as illustrated in FIG. **5**: the ACL engine **621** conducts a search through an optional look-up table, and then through rules, as illustrated in FIG. **4**, to determine whether a given packet matches a rule in the set and take action on that basis. The rules use a set of matching criteria—for example, source and destination IP address, and port number, indicating the application. These rules are fixed and use known matching criteria. The ACL engine **621** then allows some packets **622**, and denies or drops, others **623**.

As shown in FIG. **6**, in a preferred embodiment, the IP packets **620** enter the ACL engine **621**. As in the prior art, the ACL engine **621** conducts a search, using fixed rules. The ACL engine then outputs allowed packets **632**, and initially denied packets **633**.

Unlike the prior art, the firewall engine that embodies one aspect of the present invention includes additional dynamic filtering, which further processes the packets. In particular, the initially denied packets **633** are processed by a dynamic filter **637**, which allows some of the initially denied packets to pass through the firewall and enter the private network. The dynamic filter **637** conducts a search through an additional set of rules, which are dynamically generated. The dynamic filter **637** generates rules using criteria such as port number and IP address, which are extracted from incoming packets for applications, such as RealAudio, Netmeeting (which uses the H3232 protocol) and network file system (NFS).

For example, when an FTP is initiated, the first sequence of FTP packets, which includes information on the port number and the IP address, will be passed by the rules in the ACL engine **621**. The dynamic filter **637** then extracts port number and IP address from this first sequence of packets, and generates new rules, similar to the fixed rules used by the ACL, including these criteria. Later sequences of FTP packets will be denied by the ACL engine **621**, but the dynamic filter **637** will pass the packets based on the new, dynamically-generated rules. The way in which the search through the dynamically-generated rules is conducted is similar to the approach used in the ACL engine **621**. The dynamic filter then drops packets which are finally denied **636**, and allows other initially denied packets, which meet the additional access control requirements, to pass **635** through the firewall and enter the private network.

This approach to processing the incoming IP packets has advantages over the prior art. Using dynamically-generated rules allows for more flexible access policy. However, if dynamic rule generation was included in the ACL engine **621**, the processing speed would be decreased. The dynamic filter **637** used in accordance with the present invention, following the ACL engine **621**, advantageously allows the use of dynamically-generated rules, without increasing the processing time for those IP packets, which are initially allowed **632** by the ACL engine **621** based on the fixed rule set.

Another preferred embodiment, as shown in FIG. **6**, additionally allows for network address translation (NAT), to enable IP addresses, port numbers and message authentication codes (MACs) in the private network to be concealed from the public network. The public network can only access this information for the firewall. Thus, the destination information in the headers in the incoming packets must be changed, to reflect the private network IP addresses, port numbers and MAC. Furthermore, source information in the headers of outgoing packets must also be changed, to reflect the firewall network IP address, port number and MAC.

However, depending on the particular application used, information relating to the IP address or port number may be embedded in the packet content or payload, as well as in the header. In that case, the packet payload for an incoming packet must be translated to reflect the internal IP address and port number, as shown in FIG. **7***a*. Likewise, the packet payload for an outgoing packet must be translated to reflect the firewall address and port number, as shown in FIG. **7***b*.

As shown in FIG. **6**, the dynamic analyzer **638** examines those packets which are initially allowed **632** by the ACL engine **621**. The dynamic analyzer **638** determines whether a given packet may require modification, due to embedded address or port number information. The dynamic analyzer **638** then separates packets which may require modification **640** from packets which do not require modification **639**. Packets which include IP address or port number information are identified by reading a protocol-specific field in the header. The dynamic analyzer **638** allows those initially allowed packets **632** and **635** which do not require modification **639** to pass through the firewall **642** into the private network.

The packets **640** which may require modification are then passed to an application-specific handler **641**. The application-specific handler **641**, as its name suggests, processes packets **640** for a particular application, such as FTP or NFS. The application-specific handler examines the protocol, session, port number and IP address, as well as the payload. In one embodiment, the application-specific handler may modify certain packets, which have been allowed **632** and **635**. If the IP address or port number in the packet header have been changed, for an incoming packet, or must be changed, for an outgoing packet, the application-specific handler translates the payload to reflect the change. In another embodiment, multiple application-specific handlers **641** may be provided, to process packets for different applications. For example, the firewall may include both an FTP-specific handler and an NFS-specific handler.

In another embodiment, the application-specific handler **641** may include the capability to send a "reset" packet to inform the TCP processor sending the denied packets that the connection has been denied. The connection is thereby rejected, rather than merely dropped. The rejection will prevent the TCP processor sending the denied packets **636** from continuing to try to connect with the network, thereby avoiding wasted bandwidth.

In conjunction with the software functionality description provided in the present disclosure, an apparatus in accordance with the preferred embodiments may be programmed using methods known in the art as described, for example, in Francise et. al., *Professional Active Server Pages* 2.0, Wrox Press (1998), and Zaration, *Microsoft C*++6.0 *Programmer's Guide*, Microsoft Press (1998), the contents of each of which is hereby incorporated by reference into the present application.

While preferred embodiments of the invention have been described, these descriptions are merely illustrative and are not intended to limit the present invention. For example, while the preferred embodiment discusses primarily a hardware implementation of a firewall, the scope of the preferred embodiments is not so limited. Those skilled in the art will recognize that the disclosed software and methods are readily adaptable for broader network analysis applications.

What is claimed is:

1. An apparatus comprising:

a firewall engine including:

a first engine including a first set of rules for sorting incoming IP packets into initially allowed packets and initially denied packets; and

a filter including a second set of rules for receiving and further sorting the initially denied packets into allowed packets and denied packets.

2. The apparatus of claim **1**, wherein the filter dynamically generates the second set of rules.

3. The apparatus of claim **2**, wherein the first set of rules comprises fixed rules.

4. The apparatus of claim **3**, further comprising:

a second engine for receiving and further processing the initially allowed packets.

5. The apparatus of claim **4**, wherein the second engine is capable of modifying some subset of the initially allowed packets.

6. The apparatus of claim **5**, wherein the second engine comprises:

a dynamic analyzer for identifying initially allowed packets requiring network address translation; and

a handler for providing network address translation.

7. The apparatus of claim **5**, wherein the second engine comprises a dynamic analyzer for sending a "reset" packet to a source IP address.

8. A computer software product, tangibly stored on a computer-readable medium, for providing a network security, comprising instructions operable to cause a programmable processor to:

process incoming IP packets into initially allowed packets and initially denied packets;

extract matching criteria from incoming IP packets;

dynamically generate rules using the extracted matching criteria; and

further process the initially denied packets using the dynamically-generated rules.

9. The computer software product of claim **8**, wherein the instructions to process incoming IP packets use fixed rules.

10. The computer software product of claim **9**, further comprising instructions to:

further process the initially allowed packets into allowed packets and packets requiring modification.

11. The computer software product of claim **10**, further comprising instructions to:

modify control packets.

12. The computer software product of claim **11**, wherein the instructions to modify control packets include instructions for network address translation.

13. The computer software product of claim **10**, further comprising instructions to:

generate and transmit a "reset" packet in response to a denied packet.

14. A method for providing network computer security, comprising:

receiving incoming packets at a firewall;

sorting the incoming packets into initially allowed packets and initially denied packets; and

further sorting the initially denied packets into allowed and denied packets using rules.

15. The method of claim **14**, wherein the step of sorting the incoming packets is performed using fixed rules.

16. The method of claim **15**, further comprising the step of further sorting the initially allowed packets into allowed packets and packets requiring modification.

17. The method of claim **16**, further comprising the step of providing network address translation for packets requiring modification.

18. The method of claim **14**, wherein the packets are IP packets.

19. The method of claim **14**, wherein the rules are dynamically generated.

20. A method for providing network computer security, comprising:

receiving incoming IP packets at a firewall;

sorting the incoming IP packets into initially allowed packets and initially denied packets using a set of fixed rules;

extracting parameters from the incoming IP packets;

using the extracted parameters to generate a set of dynamically-generated rules; and

further sorting the initially denied packets into allowed and denied packets using the dynamically-generated rules.

21. The method of claim **20**, further comprising the step of further sorting the initially allowed packets into allowed packets and packets requiring modification.

22. The method of claim **21**, further comprising the step of providing network address translation for packets requiring modification.

23. An apparatus comprising:

an ASIC including a firewall engine including:

a first engine including a first set of rules for processing incoming IP packets into initially allowed packets and initially denied packets; and

a filter including a second set of rules for receiving and further processing the initially denied packets into allowed packets and denied packets.

24. A method for providing network computer security, comprising:

receiving incoming packets at a firewall;

processing the incoming packets into initially allowed packets and initially denied packets; and

further processing the initially denied packets into allowed and denied packets using rules.

* * * * *

EXHIBIT F

US007734752B2

(12) United States Patent
Zuk et al.

(10) **Patent No.: US 7,734,752 B2**
(45) **Date of Patent: Jun. 8, 2010**

(54) **INTELLIGENT INTEGRATED NETWORK SECURITY DEVICE FOR HIGH-AVAILABILITY APPLICATIONS**

(75) Inventors: **Nir Zuk**, Palo Alto, CA (US); **Yu Ming Mao**, Milpitas, CA (US); **Kowsik Guruswamy**, Sunnyvale, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1616 days.

(21) Appl. No.: **10/961,075**

(22) Filed: **Oct. 12, 2004**

(65) **Prior Publication Data**

US 2006/0005231 A1 Jan. 5, 2006

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/402,920, filed on Mar. 28, 2003, which is a continuation-in-part of application No. 10/072,683, filed on Feb. 8, 2002.

(51) **Int. Cl.**
G06F 15/16 (2006.01)
G06F 15/173 (2006.01)
G06F 11/00 (2006.01)

(52) **U.S. Cl.** **709/223**; 709/205; 714/11

(58) **Field of Classification Search** 709/205, 709/223; 714/11
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,598,410	A	1/1997	Stone	370/469
5,606,668	A	2/1997	Shwed	395/200.11
5,835,726	A	11/1998	Shwed et al.	395/200.59
6,006,264	A	12/1999	Colby et al.	709/226
6,052,788	A	4/2000	Wesinger, Jr. et al.	
6,119,236	A	9/2000	Shipley	713/207
6,205,551	B1	3/2001	Grosse	
6,253,321	B1	6/2001	Nikander et al.	713/160
6,275,942	B1	8/2001	Bernhard et al.	713/201
6,279,113	B1	8/2001	Vaidya	713/201
6,301,668	B1	10/2001	Gleichauf et al.	713/201

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1 143 660	10/2001

(Continued)

OTHER PUBLICATIONS

Co-pending U.S. Appl. No. 10/072,683, filed Feb. 8, 2002, entitled "Multi-Method Gateway-Based Network Security Systems and Methods," Nir Zuk et al., 62 page specification, 16 sheets of drawings.

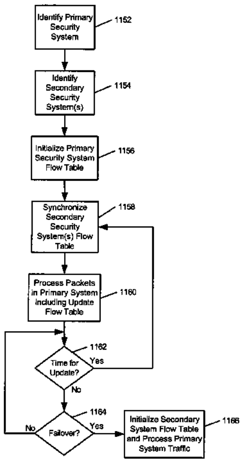
(Continued)

Primary Examiner—Rupal D Dharia
Assistant Examiner—Tanim Hossain
(74) *Attorney, Agent, or Firm*—Harrity & Harrity, LLP

(57) **ABSTRACT**

Methods and apparatuses for inspecting packets are provided. A primary security system may be configured for processing packets. The primary security system may be operable to maintain flow information for a group of devices to facilitate processing of the packets. A secondary security system may be designated for processing packets upon a failover event. Flow records may be shared from the primary security system with the secondary security system.

23 Claims, 13 Drawing Sheets


Identify Primary Security System
1152
Identify Secondary Security System(s)
1154
Initialize Primary Security System Flow Table
1156
Synchronize Secondary Security System(s) Flow Table
1158
Process Packets in Primary System Including Update Flow Table
1160
1162
Time for Update?
Yes
No
1164
No
Failover?
Yes
Initialize Secondary System Flow Table and Process Primary System Traffic
1166

U.S. PATENT DOCUMENTS

6,304,975	B1	10/2001	Shipley	713/201
6,311,278	B1	10/2001	Raanan et al.	713/201
6,321,338	B1	11/2001	Porras et al.	713/201
6,421,730	B1	7/2002	Narad et al.	709/236
6,449,647	B1	9/2002	Colby et al.	709/226
6,453,345	B2	9/2002	Trcka et al.	709/224
6,466,985	B1	10/2002	Goyal et al.	709/238
6,487,666	B1	11/2002	Shanklin et al.	713/201
6,499,107	B1	12/2002	Gleichauf et al.	713/201
6,704,278	B1 *	3/2004	Albert et al.	370/216
6,768,738	B1	7/2004	Yazaki et al.	
6,788,648	B1	9/2004	Peterson	370/252
6,851,061	B1	2/2005	Holland et al.	726/23
6,856,991	B1	2/2005	Srivastava	707/10
6,981,158	B1	12/2005	Sanchez et al.	702/188
7,006,443	B2	2/2006	Storr	370/236.1
7,376,085	B2	5/2008	Yazaki et al.	
2002/0032797	A1	3/2002	Xu	709/238
2002/0124187	A1	9/2002	Lyle et al.	713/201
2002/0161839	A1 *	10/2002	Colasurdo et al.	709/204
2003/0105976	A1	6/2003	Copeland	713/201
2003/0145225	A1	7/2003	Bruton et al.	713/201
2003/0149887	A1	8/2003	Yadav	713/200
2003/0149888	A1	8/2003	Yadav	713/200
2004/0030927	A1	2/2004	Zuk	713/201
2005/0198335	A1 *	9/2005	Brown et al.	709/229

FOREIGN PATENT DOCUMENTS

EP	1 427 162		6/2004
JP	10-107795		4/1998
JP	11-316677		11/1999
JP	2000-312225		11/2000
JP	2001-313640		11/2001
JP	2003-78549		3/2003
WO	WO 03/025766	A1	3/2003
WO	03/061238		7/2003

OTHER PUBLICATIONS

Stonesoft, "StoneBeat" '*Security Cluster White Paper*,' Aug. 2000, Finland, pp. 1-9.

Stonesoft, StoneBeat, '*Secure Highly Available Enterprise-A White Paper*,' Feb. 2001, Finland, pp. 1-10.

Stonesoft, '*StoneGate White Paper*,' Mar. 2001, Finland, pp. 1-6.

Stonesoft Corp. '*StoneGate,' product webpage*, www.stonesoft.com/document/363.html, Mar. 27, 2001 (print date), pp. 1-2.

Stonesoft Corp. '*Next Level of Network Accessibility*' webpage, www.stonesoft.com/document/183.html, Mar. 27, 2001 (print date), p. 1.

Stonesoft Corp., '*Platforms*,' webpage, www.stonesoft.com/document/186.html, Mar. 27, 2001 (print date), p. 1.

Nokia, '*Technical White Paper: The IP Clustering Power of Nokia VPN-Keeping Customers Connected*,' Apr. 2001, pp. 1-13.

Nokia, '*Nokia VPN Solutions—Nokia VPN CC2500 Gateway*,' Jan. 1, 2001, product information, pp. 1-2.

Nokia, '*Nokia VPN Solutions—Nokia VPN CC5200 Gateway*,' 2001, product information, pp. 1-2.

Nokia, *Nokia VPN Solutions—Nokia VPN CC5205 Gateway*, 2001, product information, pp. 1-2.

Axelsson, S., "Intrusion Detection Systems: A Survey and Taxonomy," *Dept. of Computer Eng*., Chalmers Univ. Of Technology, Goteborg, Sweden, Mar. 14, 2000, pp. 1-27.

Avolio, F., "Firewalls and Virtual Private Networks," CSI Firewall Archives, printed Nov. 13, 2001 URL: http://www.spirit.com/CSI/Papers/fw_+_ vpns.html, pp. 1-7.

Bace, R., "An Introduction to Intrusion Detection & Assessment," ICSA Intrusion Detection Systems Consortium White Paper, 1999, URL: http://www.icsalabs.com/html/communities/ids/whitepaper/Intrusion1.pdf, pp. 1-38.

Business Wire, Inc., "NetScreen and OneSecure Unite to Deliver Industry's First Total Managed Security Services Platform," San Jose, CA, Feb. 20, 2001, pp. 1-2.

Business Wire, Inc., "OneSecure Launches the First Co-Managed Security Services Platform," Denver, CO, Jan. 29, 2001, pp. 1-2.

Carr, Jim, "Intrusion Detection Systems: Back to Front?," *Network Magazine*, Sep. 5, 2001, URL: http://www.networkmagazine.com/article/NMG20010823S0007/2, pp. 1-9.

Check Point Software Technologies Ltd., Firewall-1® Technical Overview P/N 500326, www.checkpoint.com, Oct. 2000, pp. 1-29.

Cisco Systems, "Cisco IOS Firewall Intrusion Detection System," Cisco IOS Release 12.0(5)T, 2001, pp. 1-40.

Cisco Systems, "Cisco IOS Firewall Authentication Proxy," Cisco IOS Release 12.0(5)T, 2001, pp. 1-48.

Clark, D., "RFC815-IP Datagram Reassembly Algorithms," Internet RFC/STD/FYI/BCP Archives, http://www.faqs.org/rfcs/rfc815.html, Jul. 1982, pp. 1-8.

Copeland, Dr. John A., "Observing Network Traffic-Techniques to Sort Out the Good, the Bad, and the Ugly," PowerPoint Slide Presentation presented to ISSA-Atlanta, Jun. 27, 2001, pp. 1-22.

Denning, Dorothy E., "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, No. 2, Feb. 1987, 17 pages.

Farrow, Rik, "An Analysis of Current Firewall Technologies," *CSI 1997 Firewalls Matrix*, 1998, URL: http://www.spirit.com/CSI/Papers/farrowpa.htm, pp. 1-5.

Firewall Product Comparison Table: VelociRaptor, BorderWare Firewall Server and Firewall-1/VPN-1 Gateway, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: PIX Firewall, CyberGuard Firewall for UnixWare & CyberGuard Firewall for Windows NT, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: CyberGuard Premium Appliance Firewall, InstaGate EX & BizGuardian VPN Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Server Protector 100, GNAT Box Firewall Software & Lucent Managed Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: Internet Security and Acceleration (ISA) Server 2000, NetBSD/i386 Firewall & Guardian Firewall, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: NetScreen-10 and NetScreen-100, CyberwallPLUS & BorderManager, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: Gauntlet Firewall, Barricade Classic/XL & Barricade S, www.spirit.com, printed Nov. 13, 2001, pp. 1-8.

Firewall Product Comparison Table: Sidewinder™, SecurePipe Managed Firewall Service & SnapGear, www.spirit.com, printed Nov. 13, 2001, pp. 1-7.

Firewall Product Comparison Table: SonicWALL Pro, Sunscreen Secure Net & WinRoute Pro 4.1, www.spirit.com, printed Nov. 13, 2001, pp. 1-6.

Firewall Product Comparison Table: WatchGuard Technologies, Inc. LiveSecurity System 4.6, www.spirit.com, printed Nov. 13, 2001, pp. 1-4.

Graham, R., "FAQ: Network Intrusion Detection System," www.robertgraham.com/pubs/network-intrusion-detection.html, Ver. 0.8.3, Mar. 21, 2000, pp. 1-43.

Habra, N. et al., "ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis," Proceedings of the ESORICS '92, European Symposium on Research in Computer Security, Nov. 23-25, 1992, Toulouse, Springer-Verlag, 16 pages.

ICSA Labs, "Intrusion Detection System Buyer's Guide," ICSA White Paper, 1999, pp. 1-52.

Jackson, K. et al., "Intrusion Detection System (IDS) Product Survey," *Los Alamos National Laboratory*, Los Alamos, NM, LA-UR-99/3883 Ver. 2.1, Jun. 25, 1999, pp. 1-103.

Jones, Kyle, "Introduction to Firewalls," *IT Audit.org Forum Network Management*, vol. 2, May 1, 1999, URL: http://www.itaudit.org/forum/networkmanagement/f209nm.htm, pp. 1-5.

Lancope, "The Security Benefits of a Flow-Based Intrusion Detection System," White Paper, date unknown, pp. 1-11.

LapLink, Inc., “Article #178-Introduction to Firewalls,” www.laplink.com/support/kb/article.asp?ID_=_178, Apr. 24, 2001, pp. 1-3.

McHugh, J. et al., “Defending Yourself: The Role of Intrusion Detection Systems,” *Software Engineering Institute*, IEEE Software Eng., Sep./Oct. 2000, pp. 42-51.

Network ICE Corporation, “Why Firewalls Are Not Enough,” at www.networkice.com/products/firewalls.html, 2000, pp. 1-9.

Power, R., et al., “CSI Intrusion Detection System Resource-Five Vendors Answer Some No-Nonsense Questions on IDS,” *Computer Security Alert #184*, Jul. 1998, pp. 1-8.

Power, R., “CSI Roundtable: Experts discuss present and future intrusion detection systems,” *Computer Security Journal*, vol. XIV, #1, URL: http://www.gocsi.com/roundtable.htm, 2001, pp. 1-20.

Sample, Char, et al., “Firewall and IDS Shortcomings,” SANS Network Security, Monterey, CA, Oct. 2000, pp. 1-13.

Smith, Gary, “A Brief Taxonomy of Firewalls-Great Walls of Fire,” SANS Institute’s Information Security Reading Room, May 18, 2001, URL: http://www.sans.orq/infosecFAQ/firewall/taxonomy.htm, pp. 1-21.

Spitzner, Lance, “How Stateful is Stateful Inspection? Understanding the FW-1 State Table,” http://www.enteract.com/_~1spitz/fwtable.html, Nov. 29, 2000, pp. 1-8.

Sundaram, A., “An Introduction to Intrusion Detection,” www.acm.org/crossroads/xrds2-4/intrus.html, Jan. 23, 2001, pp. 1-12.

Tyson, Jeff, “How Firewalls Work,” http://www.howstuffworks.com/firewall.htm/printable, 2001, pp. 1-7.

Xinetica, Ltd., “An Overview of Intrusion Detection Systems,” Xinetica White Paper, Nov. 12, 2001 (print date), URL: http://www.xinetica.com/tech_explained/general/ids/wp_ids.html, pp. 1-9.

Zuk, Nir, “Protect Yourself With Firewalls,” www.techtv.com, Jul. 12, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3325761,00.html, pp. 1-3.

Zuk, Nir, “How the Code Red Worm Works,” www.techtv.com, Sep. 21, 2001, URL: http://www.techtv.com/screensavers/print/0,23102,3349133,00.html, pp. 1-2.

Petersen, S., et al., “Web apps pose security threat,” ZDNet: Tech Update, Jan. 29, 2001, URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2679177,00.html, pp. 1-3.

Lancope, “StealthWatch Provides Early Detection of the Code Red Worm and its Future Variants,” www.stealthwatch.com, date unknown, pp. 1-4.

Reavis, J., “Cash and BURN,” Jun. 2001, 6 pages.

SOS Corporation, “An Introduction to Firewalls,” 1995, URL: http://www.uclan.ac.uk/facs/destech/compute/staff/haroun/FIREWALS.HTM, pp. 1-3.

Morgan, Lisa,“ Be Afraid, Be Very Afraid,” InternetWeek Intrusion Detection Systems, Jan. 3, 2001, pp. 1-6.

Mullins, Robert, “‘Cyber war’ raises security concerns,” *Silicon Valley/San Jose Business Journal*, May 11, 2001, pp. 1-4.

James P. Anderson Co., “Computer Security Threat Monitoring and Surveillance,” Apr. 15, 1980, 56 pages.

Internet Security Systems, Inc., “REALSECURE™, The RealSecure Advantage,” 2001, 2 pages.

Chuvakin, A., et al., “Basic Security Checklist for Home and Office Users,” SecurityFocus, Nov. 5, 2001, pp. 1-5.

Network Ice, “SMTP WIZ command,” 2001, URL: http://networkice.com/Advice/Intrusions/2001006/default.htm, pp. 1-2.

Bace, R., et al., “NIST Special Publication on Intrusion Detection Systems,” National Institute of Standards and Technology Special Publication, date unknown, pp. 1-51.

European Search Report for European Application No. 05 02 2282, Jan. 11, 2006. 2 pp.

Navarro, A Partial Deterministic Automaton for Approximate String Matching, 1997, Department of Computer Science, University of Chile, 13 pages.

Navarro et al., Improving an Algorithm for Approximate Pattern Matching, 1998, Department of Computer Science, University of Chile, pp. 1-34.

Network Magazine, vol. 2, No. 2, pp. 116-119 (with English abstract).

Software Design, Nov. 1996, pp. 39-58 (with English abstract).

* cited by examiner

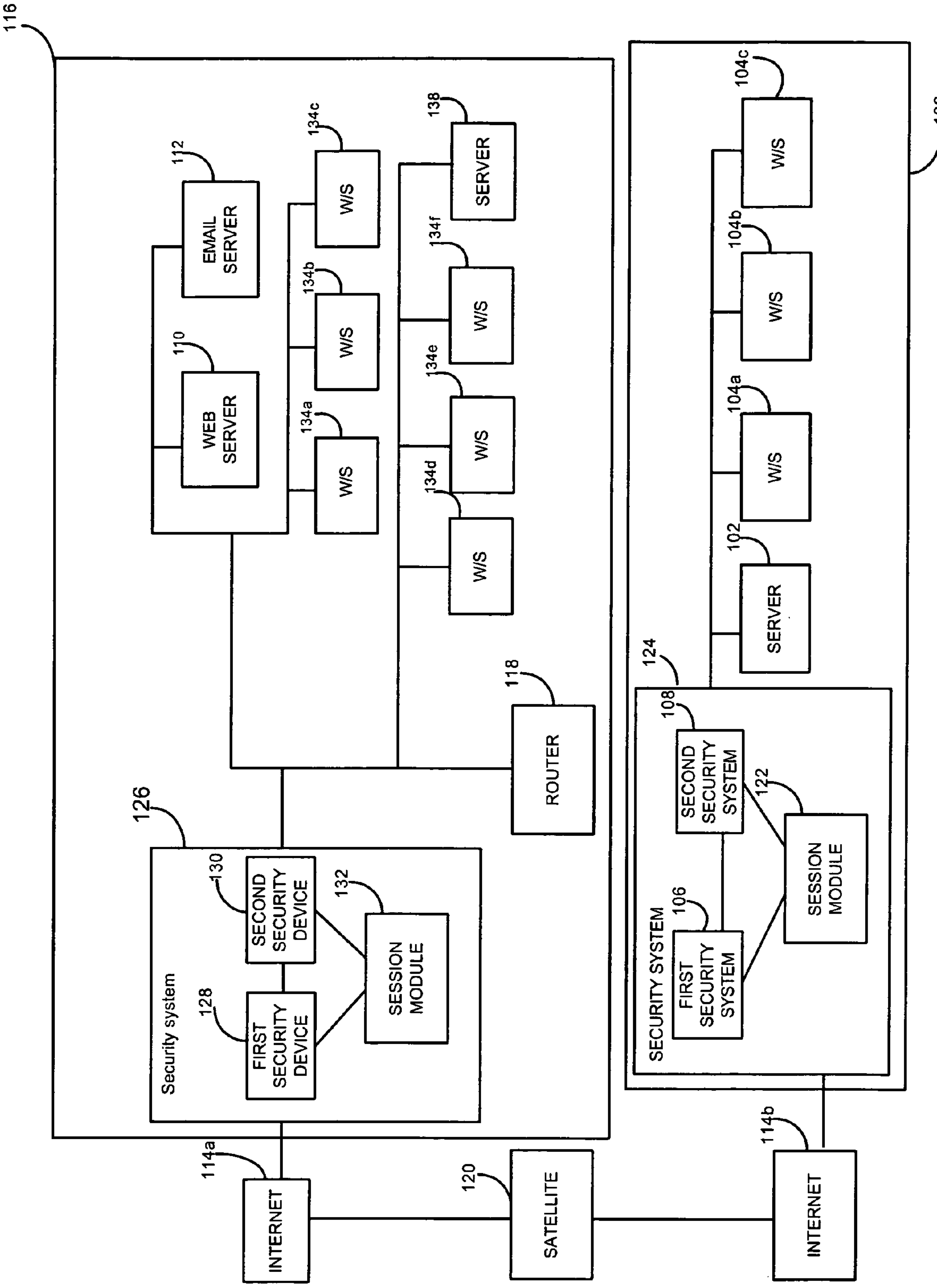

116
112
EMAIL SERVER
110
WEB SERVER
134c
W/S
134b
W/S
134a
W/S
138
SERVER
134f
W/S
134e
W/S
134d
W/S
118
ROUTER
126
Security system
128
FIRST SECURITY DEVICE
130
SECOND SECURITY DEVICE
132
SESSION MODULE
114a
INTERNET
120
SATELLITE
114b
INTERNET
100
104c
W/S
104b
W/S
104a
W/S
102
SERVER
124
SECURITY SYSTEM
106
FIRST SECURITY SYSTEM
108
SECOND SECURITY SYSTEM
122
SESSION MODULE

FIG. 1

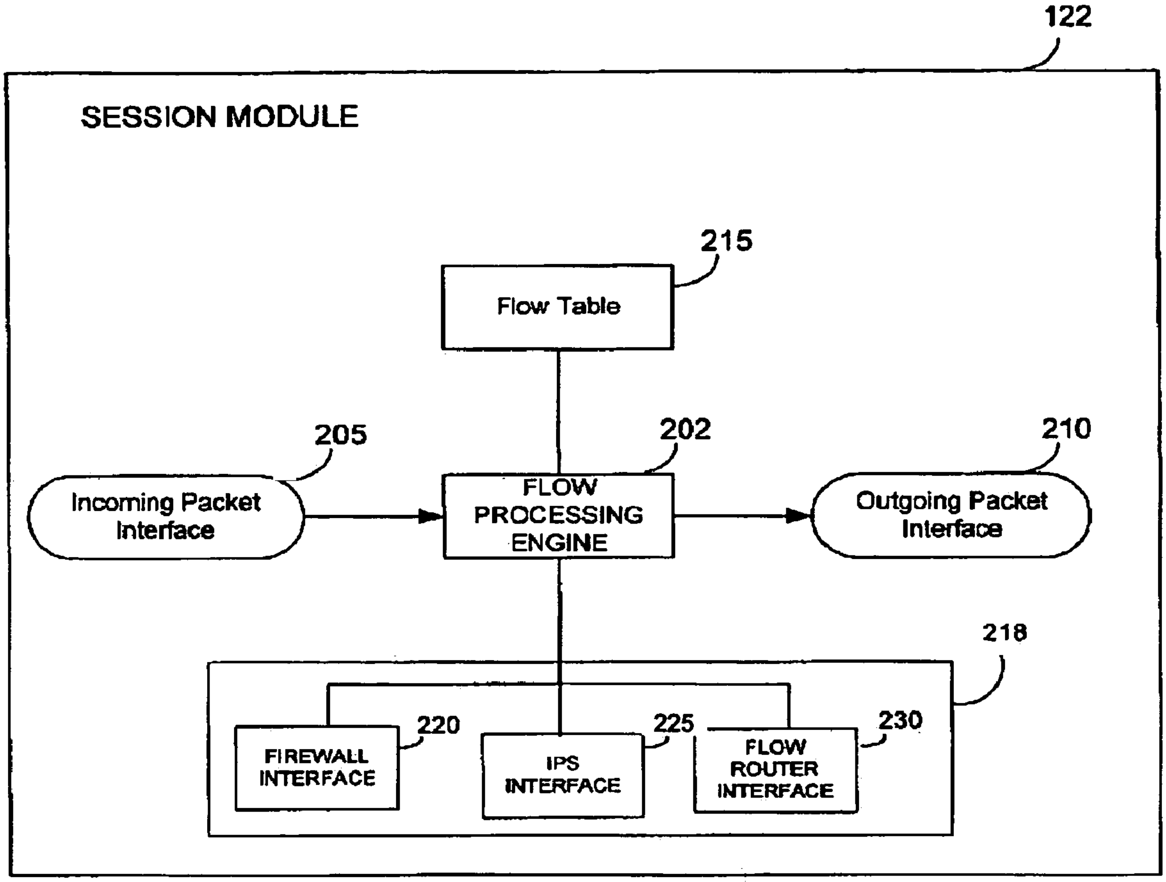
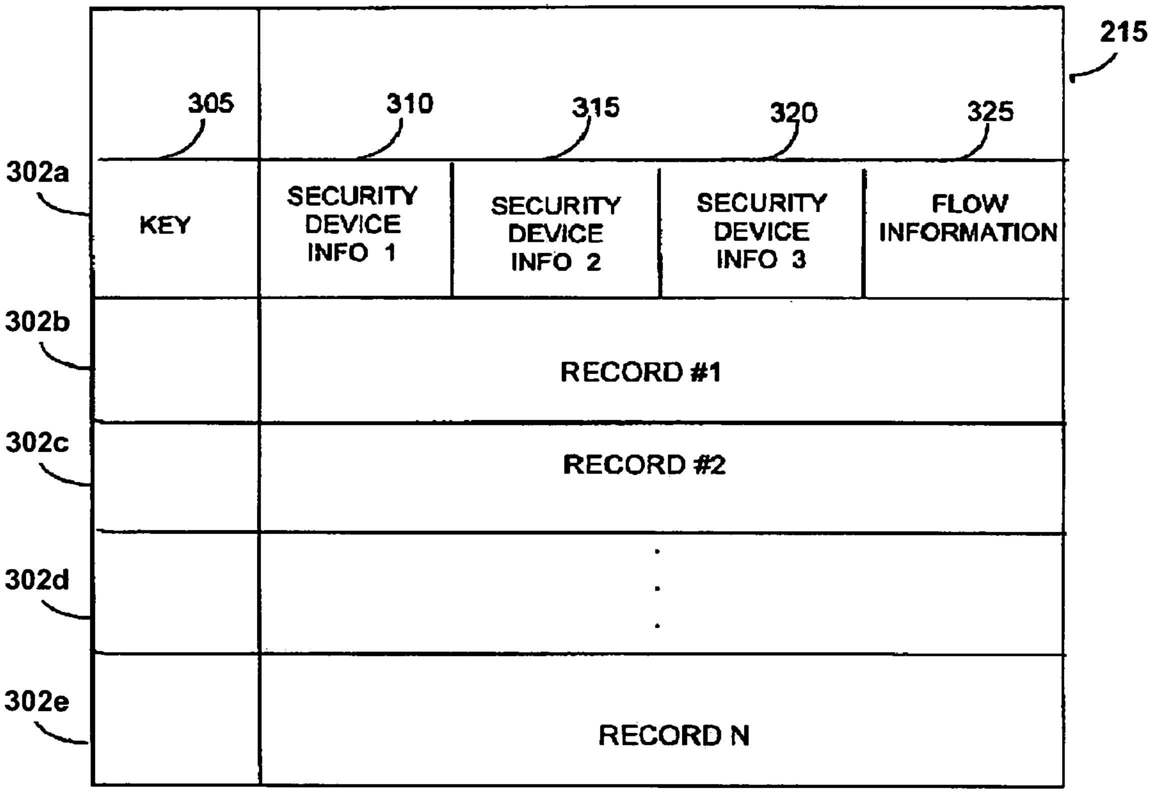

122
SESSION MODULE
215
Flow Table
205
Incoming Packet Interface
202
FLOW PROCESSING ENGINE
210
Outgoing Packet Interface
218
220
FIREWALL INTERFACE
225
IPS INTERFACE
230
FLOW ROUTER INTERFACE

FIG. 2



KEY	SECURITY DEVICE INFO 1	SECURITY DEVICE INFO 2	SECURITY DEVICE INFO 3	FLOW INFORMATION
	RECORD #1			
	RECORD #2			
	. . .			
	RECORD N			

FIG. 3

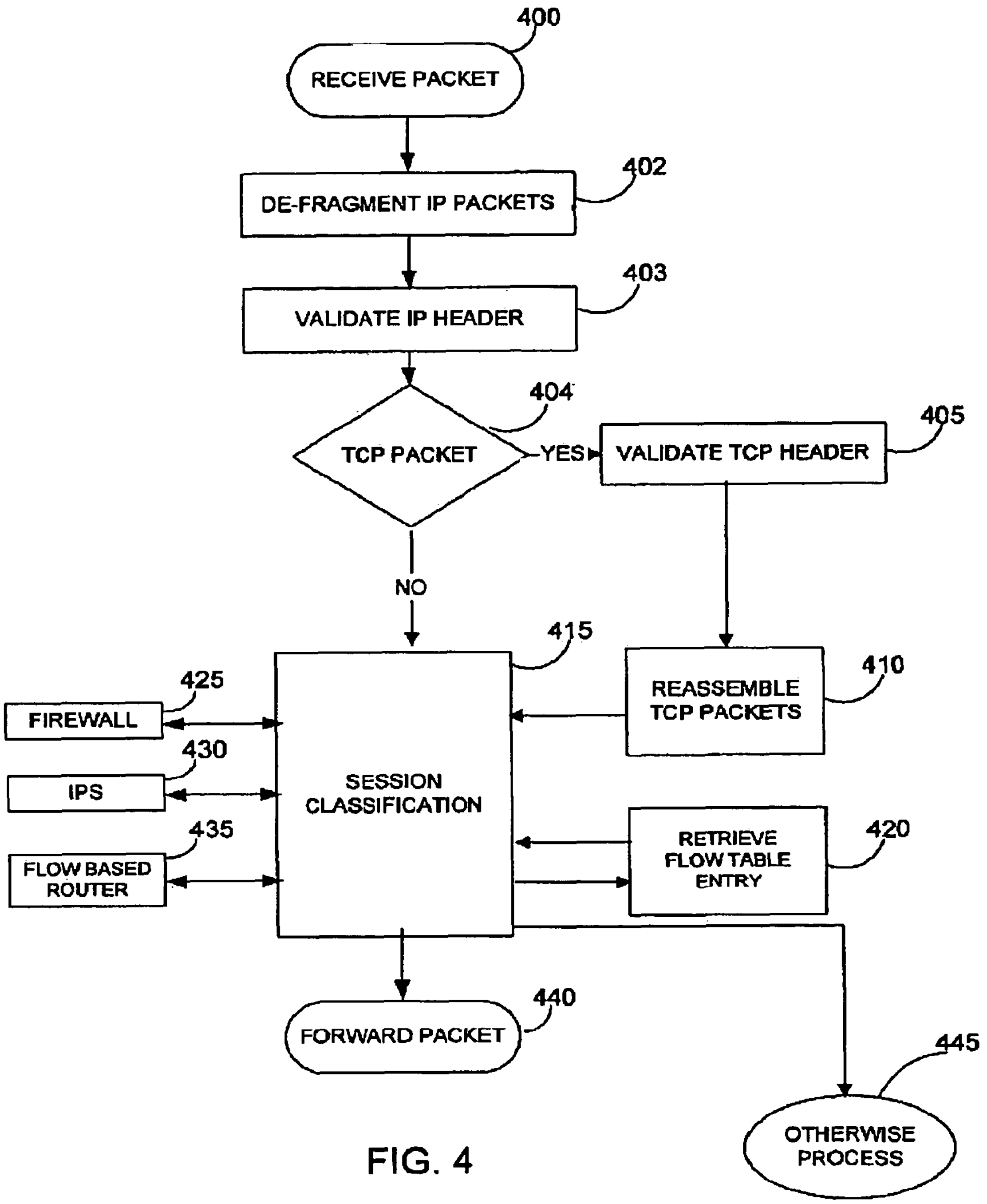

400
RECEIVE PACKET
402
DE-FRAGMENT IP PACKETS
403
VALIDATE IP HEADER
404
TCP PACKET
YES
405
VALIDATE TCP HEADER
NO
415
410
REASSEMBLE TCP PACKETS
425
FIREWALL
430
IPS
435
FLOW BASED ROUTER
SESSION CLASSIFICATION
420
RETRIEVE FLOW TABLE ENTRY
440
FORWARD PACKET
445
OTHERWISE PROCESS

FIG. 4

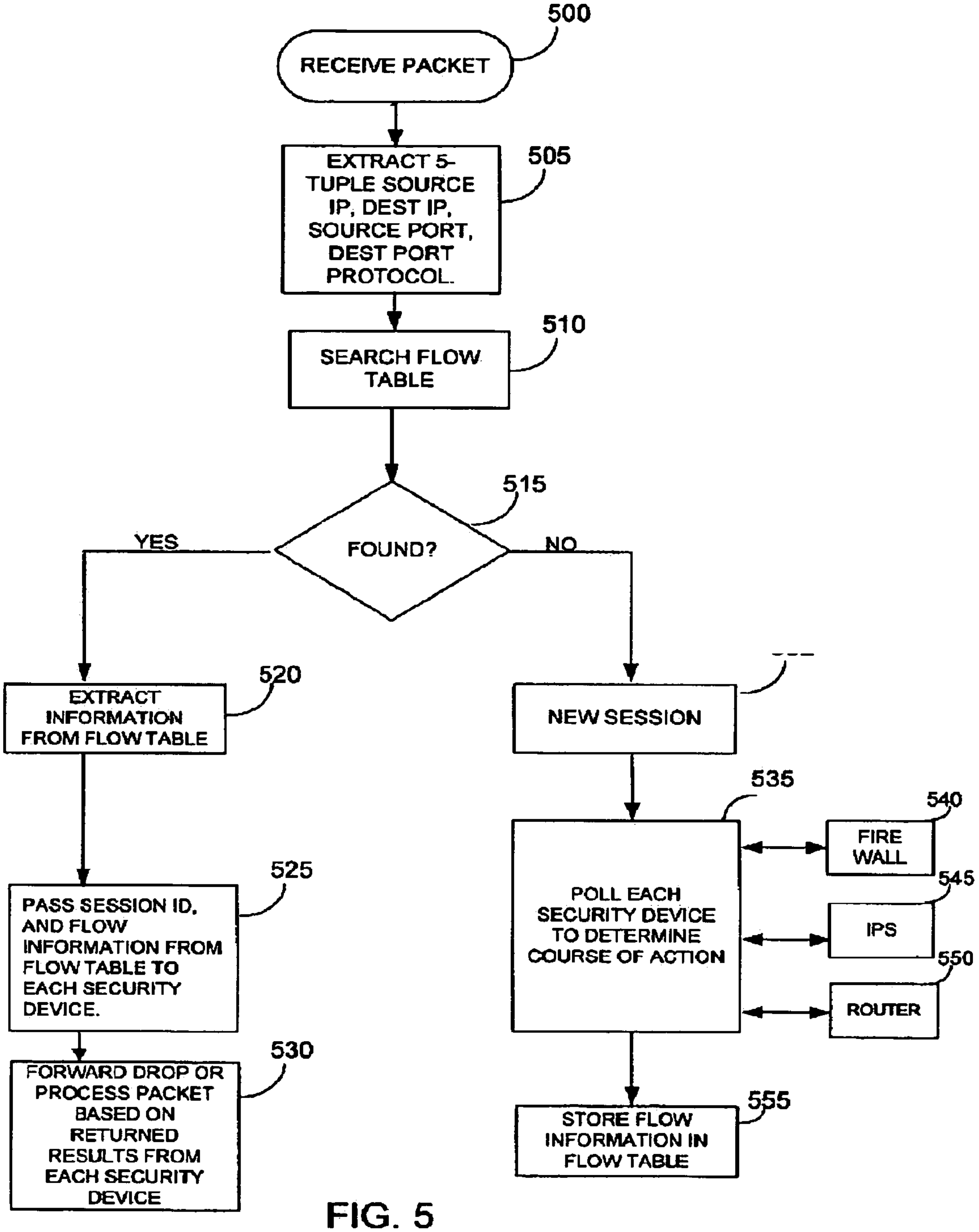

500
RECEIVE PACKET
505
EXTRACT 5-TUPLE SOURCE IP, DEST IP, SOURCE PORT, DEST PORT PROTOCOL.
510
SEARCH FLOW TABLE
515
FOUND?
YES
NO
520
EXTRACT INFORMATION FROM FLOW TABLE
NEW SESSION
525
PASS SESSION ID, AND FLOW INFORMATION FROM FLOW TABLE TO EACH SECURITY DEVICE.
535
POLL EACH SECURITY DEVICE TO DETERMINE COURSE OF ACTION
540
FIRE WALL
545
IPS
550
ROUTER
530
FORWARD DROP OR PROCESS PACKET BASED ON RETURNED RESULTS FROM EACH SECURITY DEVICE
555
STORE FLOW INFORMATION IN FLOW TABLE

FIG. 5

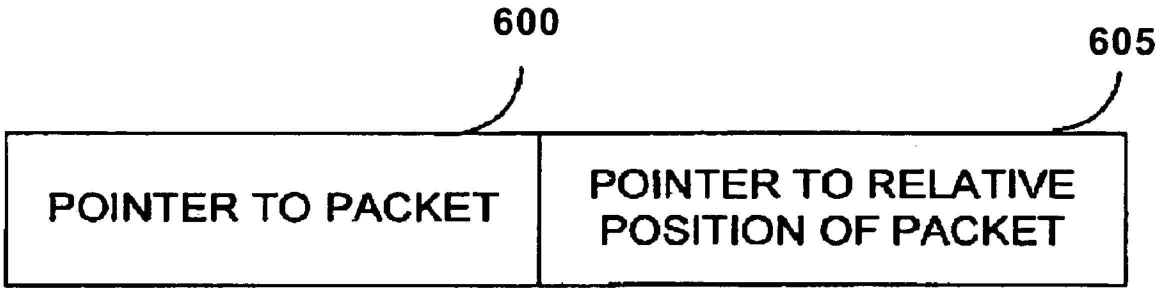

600
605
POINTER TO PACKET
POINTER TO RELATIVE POSITION OF PACKET

FIG. 6

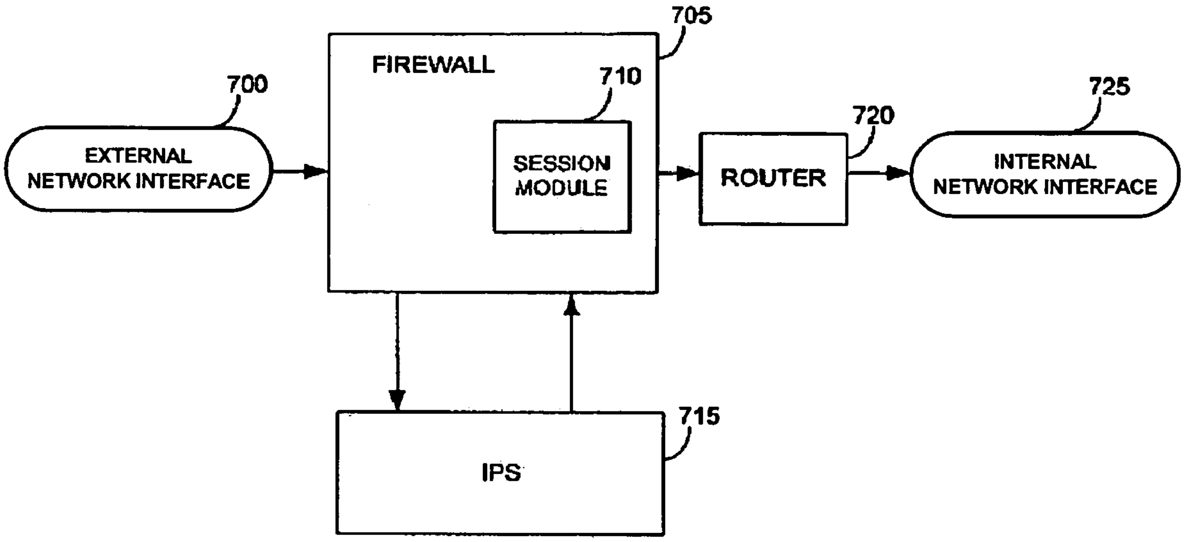

705
FIREWALL
700
710
EXTERNAL NETWORK INTERFACE
SESSION MODULE
720
ROUTER
725
INTERNAL NETWORK INTERFACE
715
IPS

FIG. 7

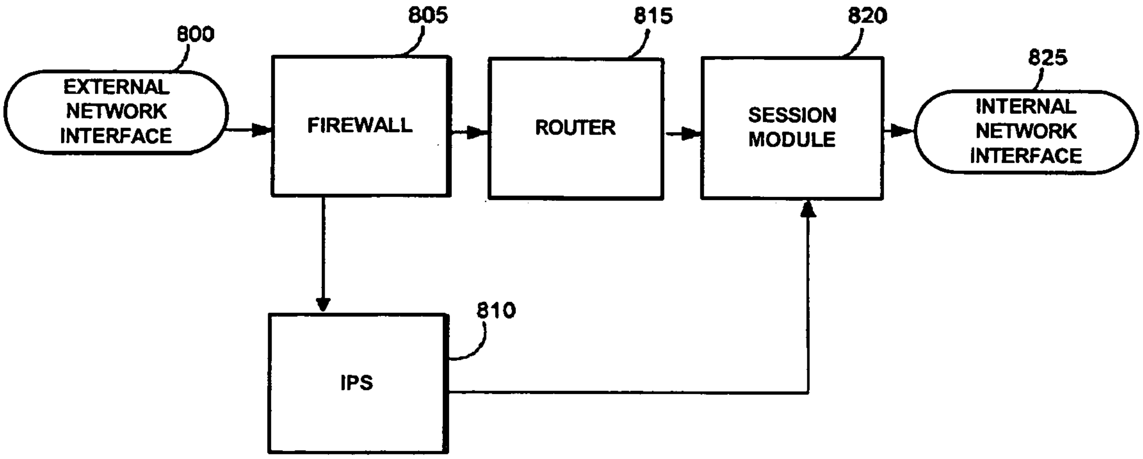

800
EXTERNAL NETWORK INTERFACE
805
FIREWALL
815
ROUTER
820
SESSION MODULE
825
INTERNAL NETWORK INTERFACE
810
IPS

FIG. 8

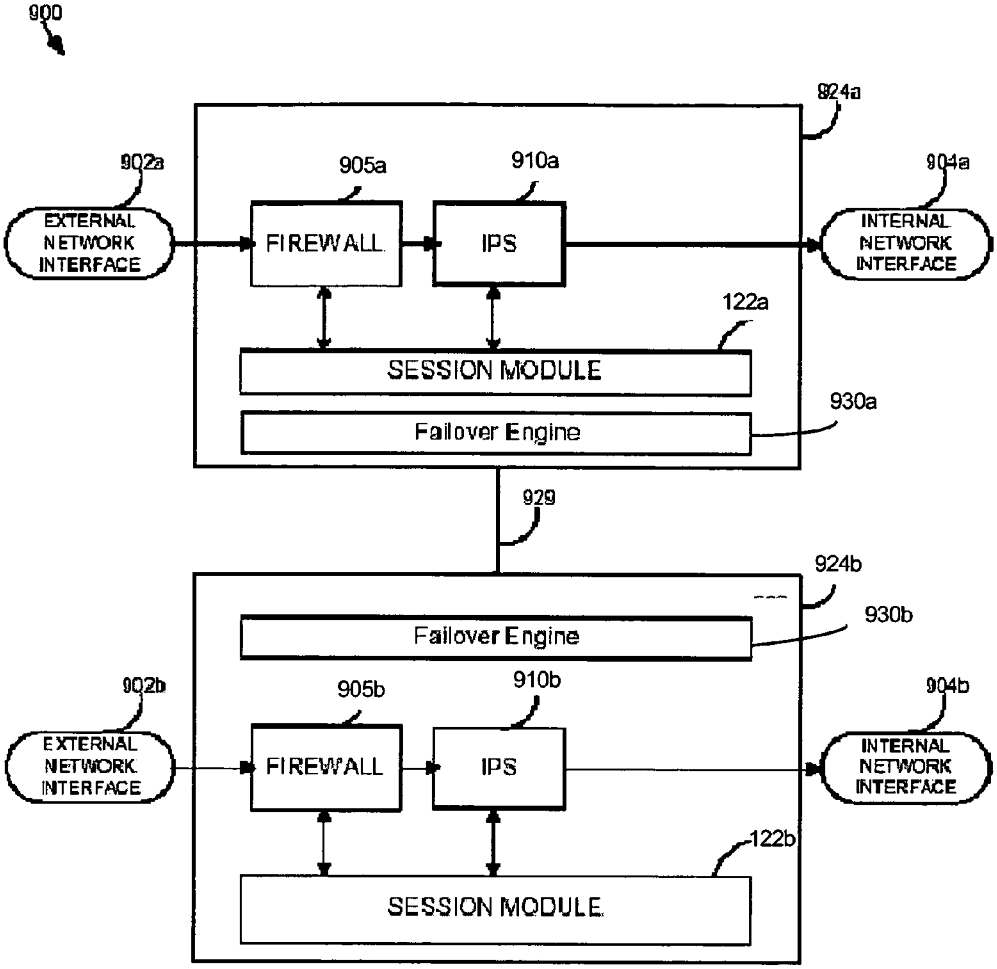

900
924a
902a
905a
910a
904a
EXTERNAL NETWORK INTERFACE
FIREWALL
IPS
INTERNAL NETWORK INTERFACE
122a
SESSION MODULE
930a
Failover Engine
929
924b
930b
Failover Engine
902b
905b
910b
904b
EXTERNAL NETWORK INTERFACE
FIREWALL
IPS
INTERNAL NETWORK INTERFACE
122b
SESSION MODULE

FIG. 9

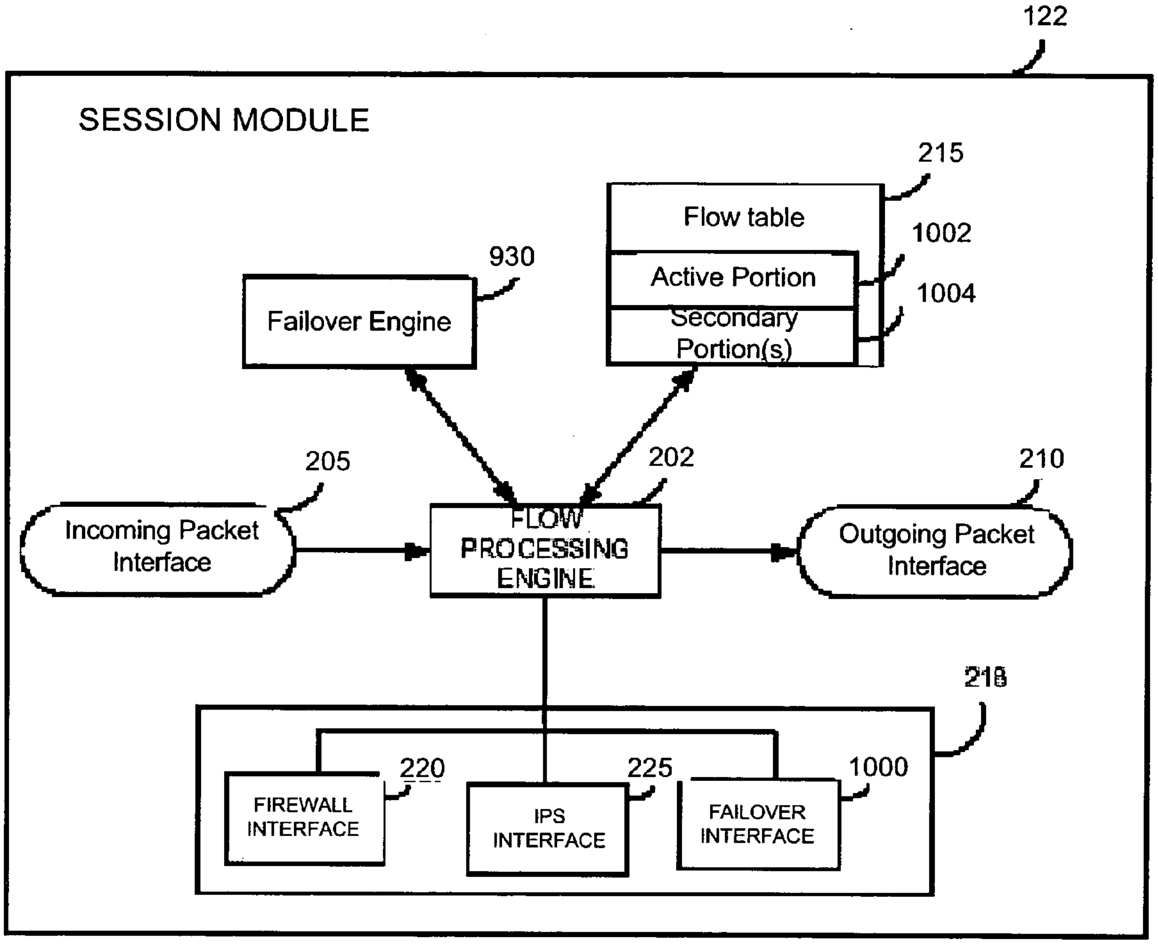

122
SESSION MODULE
215
Flow table
1002
Active Portion
1004
Secondary Portion(s)
930
Failover Engine
205
Incoming Packet Interface
202
FLOW PROCESSING ENGINE
210
Outgoing Packet Interface
218
220
FIREWALL INTERFACE
225
IPS INTERFACE
1000
FAILOVER INTERFACE

FIG. 10

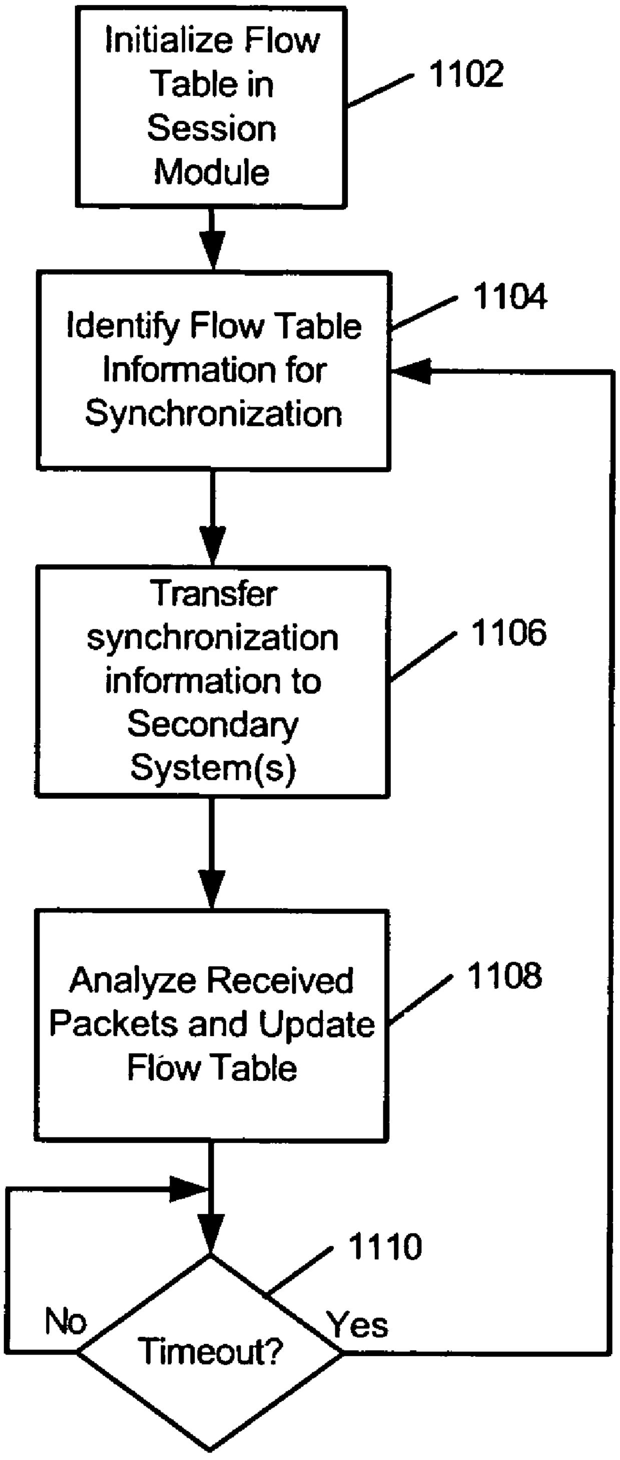

Initialize Flow Table in Session Module
1102
Identify Flow Table Information for Synchronization
1104
Transfer synchronization information to Secondary System(s)
1106
Analyze Received Packets and Update Flow Table
1108
1110
No
Yes
Timeout?

FIG. 11a

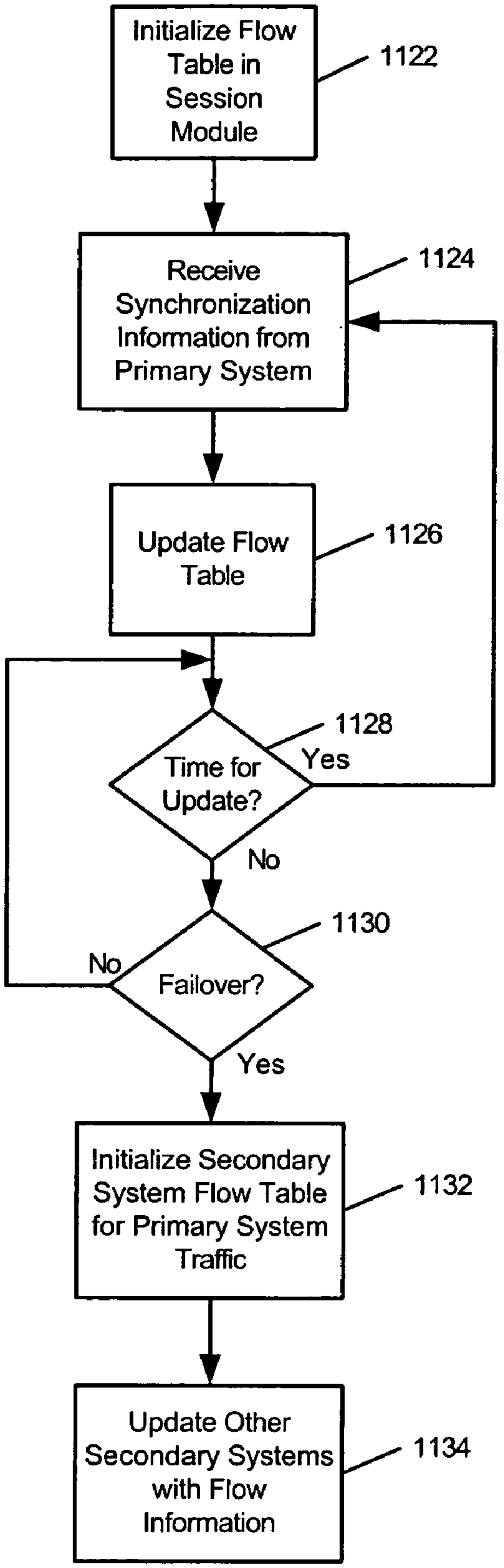

Initialize Flow Table in Session Module
1122
Receive Synchronization Information from Primary System
1124
Update Flow Table
1126
Time for Update?
1128
Yes
No
Failover?
1130
No
Yes
Initialize Secondary System Flow Table for Primary System Traffic
1132
Update Other Secondary Systems with Flow Information
1134

FIG. 11b

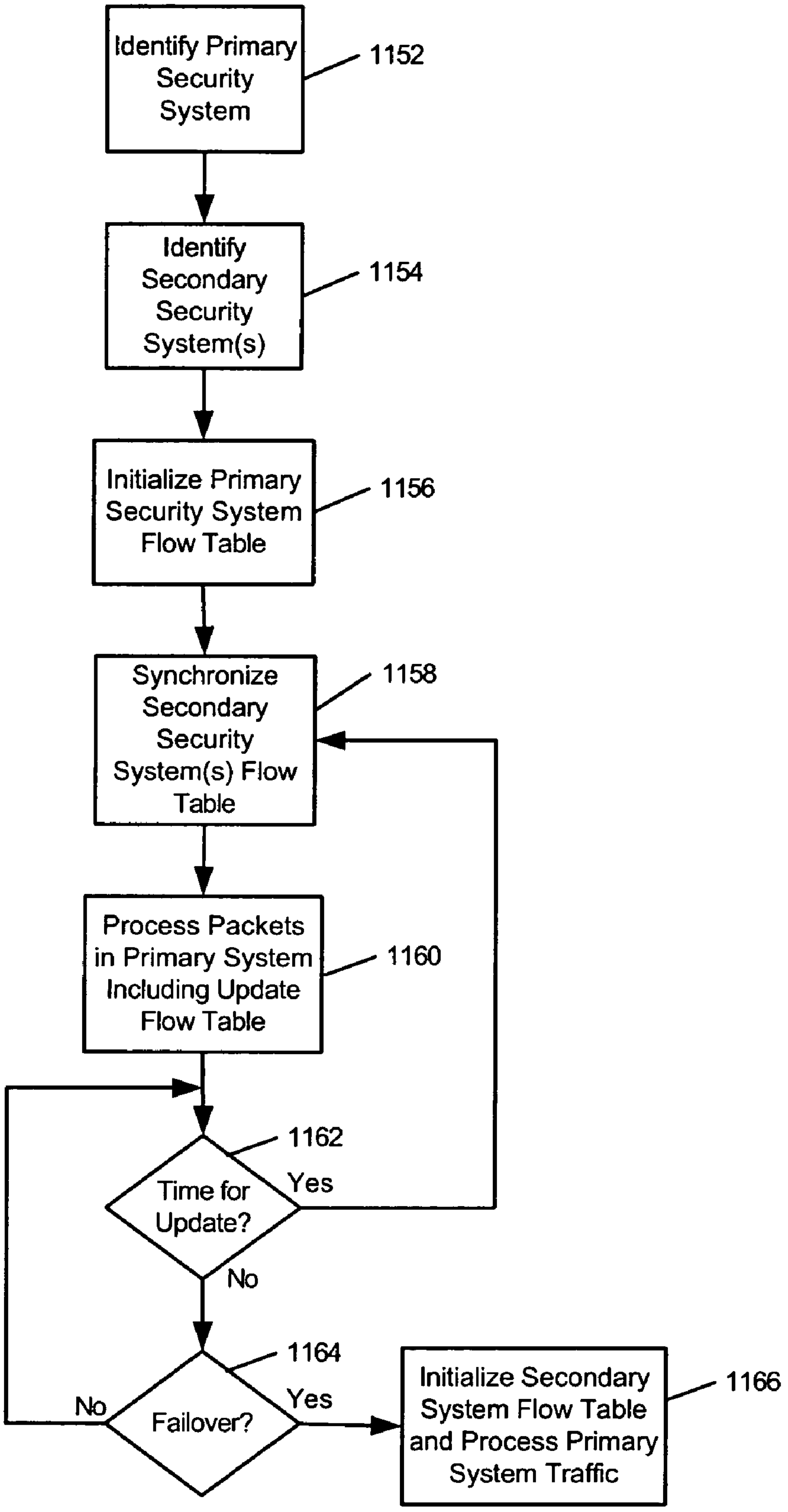

Identify Primary Security System
1152
Identify Secondary Security System(s)
1154
Initialize Primary Security System Flow Table
1156
Synchronize Secondary Security System(s) Flow Table
1158
Process Packets in Primary System Including Update Flow Table
1160
Time for Update?
1162
Yes
No
Failover?
1164
No
Yes
Initialize Secondary System Flow Table and Process Primary System Traffic
1166

FIG. 11c

INTELLIGENT INTEGRATED NETWORK SECURITY DEVICE FOR HIGH-AVAILABILITY APPLICATIONS

CROSS-REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part of U.S. patent application Ser. No. 10/402,920, filed Mar. 28, 2003, which is a continuation-in-part of U.S. patent application Ser. No. 10/072,683, filed Feb. 8, 2002. Both U.S. patent applications are herein incorporated by reference in their entirety.

FIELD OF THE INVENTION

Systems, apparatuses and methods consistent with the principles of the invention relate generally to controlling computer network security.

BACKGROUND

Firewalls and intrusion detection systems are devices that may be used to protect a computer network from unauthorized or disruptive users. A firewall can be used to secure a local area network from users outside the local area network. A firewall checks, routes, and frequently labels all messages sent to or from users outside the local area network. An intrusion detection system (IDS) can be used to examine information being communicated within a network to recognize suspicious patterns of behavior. Information obtained by the IDS can be used to block unauthorized or disruptive users from accessing the network. An intrusion prevention system (IPS) is an in-line version of an IDS. An IPS can be used to examine information as it is being communicated within a network to recognize suspicious patterns of behavior.

A flow-based router (FBR) can allow network administrators to implement packet forwarding and routing according to network policies defined by a network administrator. FBRs can allow network administrators to implement policies that selectively cause packets to be routed through specific paths in the network. FBRs can also be used to ensure that certain types of packets receive differentiated, preferential service as they are routed. Conventional routers can forward packets to their destination address based on available routing information. Instead of routing solely based on the destination address, FBRs can enable a network administrator to implement routing policies to allow or deny packets based on several other criteria including the application, the protocol, the packet size and the identity of the end system.

A packet filter can operate on the data in the network layer to defend a trusted network from attack by an untrusted network. For example, packet filters inspect fields of the Transmission Control Protocol/Internet Protocol (TCP/IP) header including, the protocol type, the source and destination Internet Protocol (IP) address, and the source and destination port numbers. Disadvantages of packet filters include, slow speed and difficult management in large networks with complex security policies.

A proxy server can operate on values carried in the application layer to insulate a trusted network from an untrusted network. In an application proxy server, two Transmission Control Protocol (TCP) connections may be established: one between the packet source and the proxy server, another between the proxy server and the packet destination. The application proxy server can receive the arriving packets on behalf of the destination server. The application data can be assembled and examined by the proxy server, and a second

TCP connection can be opened between the proxy server and the destination server to relay permitted packets to the destination server. Proxy servers can be slow because of the additional protocol stack overhead required to inspect packets at the application layer. Furthermore, because a unique proxy can be required for each application, proxy servers can be complex to implement and difficult to modify for supporting new applications. In addition, because proxy servers only examine application packets, proxy servers may not detect an attempted network security intrusion at the TCP or network layers.

SUMMARY

The present invention provides methods and apparatuses for inspecting packets.

In a first aspect, a method is provided for inspecting packets. The method may include configuring a primary security system for processing packets, where the primary security system is operable to maintain flow information for a group of devices to facilitate processing of the packets, designating a security system for processing packets upon a failover event, and sharing flow records from the primary security system with the secondary security system

In a second aspect, a system is provided. The system may include a first apparatus. The first apparatus may include a first security device, a first module operable to maintain flow information associated with packets received from a computer network, and a communication interface operable to permit an exchange of flow records with a second apparatus. The first module is further operable to share device-specific flow information with the first security device.

In a third aspect, a system for inspecting packets is provided. The system may include a primary security apparatus operable to receive and process packets. The primary security apparatus may include means for maintaining flow information for a group of devices included in the primary security apparatus. A secondary apparatus is operable to process packets for the primary security apparatus when a failover event occurs. The secondary security apparatus further includes means for sharing flow information among a group of devices. The system further comprises means for sharing flow records from the primary security apparatus to the secondary security apparatus.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

DESCRIPTION OF DRAWINGS

FIG. **1** shows a network topology including a session module.

FIG. **2** illustrates a block diagram of the session module.

FIG. **3** shows the structure of a flow table.

FIG. **4** is a flowchart describing the operation of the session module.

FIG. **5** is a flowchart describing session classification.

FIG. **6** shows the quasi-reassembly information generated by the session module.

FIG. **7** shows a network topology where the session module is included in a firewall.

FIG. **8** shows a network topology where the session module operates in series with a firewall, IPS, and router.

FIG. **9** shows a network topology where a session module, firewall, IPS and router are included in a single security device.

FIG. **10** shows a network topology where a group of security devices is included in a high availability architecture.

FIG. **11***a-c* illustrate processes for providing failover protection in the network topology of FIG. **10**.

Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

FIG. **1** shows an exemplary network topology including a local area network (LAN) **100**, including a server **102**, several workstations (W/S) **104***a*-**104***c* (collectively, "**104**"), and a security system **124**. The security system **124** may include a session module **122** and a group of other security devices. In the implementation shown, security system **124** may include two security devices, a first security device **106** and a second security device **108**. LAN **100** may be connected to an external network e.g., the Internet **114***b*, through security system **124**. LAN **100** may also be connected to a second LAN **116** through an external network, e.g., Internet **114***a*. Second LAN **116** may include a web server **110**, an email server **112**, a server **138**, several workstations **134***a*-**134***f* (collectively, "**134**") and a security system **124**. LAN **116** is connected to Internet **114***a* via security system **126**. Security system **126** may include a first security device **128**, second security device **130**, and session module **132**. The computers, servers and other devices in the LAN may be interconnected using a number of data transmission media such as, for example, wire, fiber optics, and radio waves. Security system **124** and security system **126** may operate in a similar manner. Using security system **124** by way of example, session module **122** may monitor packets being communicated within the network. In one implementation, first security device **106** may be a firewall and second security device **108** may be an IPS. Session module **122** may act in conjunction with first security device **106** and second security device **108** to facilitate blocking of packets associated with an attempted network security intrusion.

FIG. **2** shows a block diagram of an exemplary session module, such as session module **122**. Session module **122** may include an incoming packet interface **205** for receiving packets. Session module **132** may operate in a similar manner. The received packets may be analyzed by a flow processing engine (FPE) **202** to determine if an attempted network security intrusion is in progress. Session module **122** may also include a flow table **215**. Flow table **215** may be used to store information regarding flows associated with received packets. Session module **122** may also include interfaces to other security devices on the network. In one implementation, session module **122** may include a firewall interface **220**, an IPS interface **225**, and a flow-based router interface **230**. Security device interfaces **218** may be used by session module **122** to obtain information regarding the received packet, and information regarding the flow associated with the packet, in order to determine if the received packet should be allowed or modified. Security device interfaces **218** may also be used by session module **122** to communicate flow information that may be used by the security devices to facilitate processing of the packet.

FIG. **3** illustrates a structure of a flow table **215** that may be used in implementations consistent with the principles of the invention. Flow table **215** may include flow records **302***a*-**302***e* (collectively, "**302**") associated with current TCP/IP flows. A TCP/IP flow may include a sequence of data packets communicating information between a source and a destination in one direction. The flow records may be indexed using an indexing key **305**. Indexing key **305** may be used to store and retrieve the appropriate flow record associated with a received packet. In one implementation, indexing key **305** may be a hash key and flow table **215** may be implemented as a hash table. Session module **122** (FIG. **2**) may store instructions for two or more security devices on the network in the same flow record. In one implementation of session module **122**, instructions for three security devices (e.g., devices **310**, **315**, and **320**) may be stored in flow records **302**. Flow records **302** may store policy information (e.g., firewall policy, IPS policy etc., to apply to the flow) as well as other information that may be used by the security devices such as encryption parameters, address translation parameters, bookkeeping information, and statistics. Flow records **302** can also include flow information **325** for use by session module **122** in order to decide whether the packet should be allowed. Such information can include information required to implement network policies regarding, for example connection time out, time billing, and bandwidth usage. Flows, sessions and flow tables are described in greater detail in a co-pending and commonly owned patent application entitled "Multi-Method Gateway-Based Network Security Systems and Methods," and assigned U.S. patent application Ser. No. 10/072,683, the contents of which are herein incorporated by reference in its entirety.

FIG. **4** a flowchart that describes exemplary operation of FPE **202** (FIG. **2**) in an implementation consistent with the principles of the invention. Incoming packets may be received by session module **122** (act **400**). IP packets may be de-fragmented (act **402**) and the IP header may be validated for each IP packet (act **403**). During act **403**, the IP header associated with a given packet may be extracted and inspected for fundamental flaws.

If the packet is a TCP packet (act **404**), the TCP header may be validated (act **405**) and the TCP packets may be reassembled (act **410**). The validation process may include extracting TCP header data and evaluating the header for fundamental flaws. Quasi-reassembly information developed in act **410** may be communicated by session module **122** to other security devices to facilitate processing of the packet by the other security devices. Reassembly is described in greater detail below and in U.S. patent application Ser. No. 10/072,683.

In act **415**, FPE **202** may perform session classification using the TCP/IP header data associated with a given received packet. Session module **122** may determine if the packet should be allowed based on information obtained regarding the TCP/IP flow associated with the received packet and retrieved from flow table entry (act **420**). In addition, session module **122** may use information returned from one of the other security devices e.g., a firewall (act **425**), an IPS (act **430**), and a flow based router (act **435**). Further, session module **122** may also facilitate the operation of the security devices by communicating flow information to a respective device for processing a given packet. Finally, FPE **202** may forward the packet if the packet should be allowed (act **440**). Otherwise, the packet is otherwise processed (act **445**). Otherwise processing may include logging particular information regarding the packet, holding the packet, or modifying and/or dropping the packet.

FIG. **5** is a flowchart that illustrates exemplary processing that may be included in session classification (act **415** of FIG. **4**). The session classification act may receive a packet (act **500**) and extract information that may be used to determine whether the packet should be allowed. The extracted information may include source and destination IP addresses, source and destination port numbers, and a protocol (act **505**). The extracted information may be used to search flow table **215** (act **510**) in order to determine if the packet is associated with a known session flow. For a known session flow, act **510** may produce a matching flow record in flow table **215** (act **515**). If a matching flow record is found, FPE **202** (FIG. **2**) may extract TCP/IP session information for the received packet (act **520**) from the matching flow record. FPE **202** may determine whether the received packet should be allowed using the TCP/IP session information obtained during act **520**. More specifically, FPE **202** may extract information from the matching flow record, and may pass the information to the security devices (e.g., communicating the session ID and the TCP/IP session information as well as any other security device specific information from the flow record) (act **525**). Depending on the returned results from the security devices, FPE **202** can forward, drop, log, store, modify or otherwise process the given packet (act **530**).

If a matching flow record is not found in flow table **215** during act **515**, the received packet may be associated with a new TCP/IP session (act **532**). For a new TCP/IP session, FPE **202** may assign a session ID to the new session and FPE **202** may communicate with the other security devices (e.g. firewall, IPS, flow router) to determine a security policy for packets associated with the new session. For example, FPE **202** may obtain information from a firewall (act **540**) in order to determine if received packets associated with the new session should be allowed. FPE **202** may communicate with an IPS (act **545**) in order to determine if the received packet should be blocked because it matches known attack signatures for attempted network security intrusions. FPE **202** may obtain any network policy associated with the new session from a flow router (act **550**). FPE **202** may act as an arbiter between the different security devices and use the information obtained from the security devices either individually or in combination to determine if the packets associated with the new TCP/IP session should be allowed. FPE **202** may use the information obtained from the security devices to create a new flow record and may store the new flow record in flow table **215** (act **555**). The new flow record may include the TCP/IP session information for the new session associated with the received packet and any other specific security device information. Thereafter, FPE **202** may facilitate the processing of received packets associated with a given TCP/IP session as described above in association with FIG. **4** including communicating the session ID, TCP/IP session information and security device specific information to the security devices from a corresponding flow record.

In addition to determining if a received packet is associated with an attempted network security intrusion using the varied security devices, session module **122** (FIG. **2**) may also perform quasi-reassembly of the received TCP/IP packets as described above in association with FIG. **4**. FIG. **6** shows exemplary quasi-reassembly information that may be generated by session module **122**. The quasi-reassembly information may include a pointer to a location of a given packet **600** in memory and a pointer to information including a relative position of the packet in a flow **605**. In one implementation, an IPS may perform passive TCP/IP reassembly and pointer to the location of the packet **600** may be used to locate the packet within the IPS. In another implementation, pointer to information containing the relative position of the packet in the flow **605** may be used to obtain the TCP/IP sequence number included in the TCP/IP header associated with the packet. The quasi-reassembly information may be communicated to the security devices connected to session module **122** (FIG. **2**). The security devices may use the quasi-reassembly information to process the received packet.

Session module **122**, described above, may be used in a number of different network topologies. FIG. **7** shows a network topology where a session module **122** is integrated into a firewall **705**. Firewall **705** may include an interface to a router **720** and an IPS **715**. Firewall **705** may receive packets from an external network interface **700**. Firewall **705** may communicate with IPS **715** to determine whether the received packet should be blocked based on known attack signatures. If firewall **705** and IPS **715** determine that the packet should be allowed to pass, firewall **705** may send the received packet to router **720**. Router **720** may forward the outgoing packet to its intended destination, using an internal network interface **725**, based on the network policies stored in the router.

FIG. **8** shows an exemplary alternate arrangement for implementing computer network security using session module **122**. In this arrangement, session module **820** may operate in series with a firewall **805**, an IPS **810**, and a router **815**. Packets received using an external network interface **800** may be screened by firewall **805** before being communicated to router **815**. Firewall **805** may also send information regarding the received packet to IPS **810**. IPS **810** may examine the received packet and may inform session module **820** if the received packet should be blocked based on known attack signatures. Router **815** may send the packet to session module **820** for further processing. If session module **820** determines that the received packet should be allowed it may forward the received packet to its intended destination using an internal network interface **825**.

FIG. **9** shows an exemplary high availability arrangement for implementing computer network security using session module **122**. In this arrangement, a network topology may include a local area network (LAN) **900**, including external network interfaces **902***a*, internal network interfaces **904***a*, and a first security system **924***a*. First security system **924***a* may include session module **122***a* and a group of other security devices. In the implementation shown, first security system **924***a* includes two security devices, a firewall device **905***a* and an IPS device **910***a*. In other implementations, first security system **924***a* may include more or fewer security devices, such as, for example, one firewall **905***a* and no IPS. LAN **900** may be connected through first security system **924***a* to an external network e.g., the Internet, by external network interface **902***a*. LAN **900** may also be connected through a second security system **924***b* to an external network e.g., the Internet, by external network interface **902***b*. Second security system **924***b* may include session module **122***b* and a group of security modules. In the implementation shown, second security system **924***b* may include two security devices, a firewall **905***b* and an IPS device **910***b* and an internal network interface **904***b*. In other implementations, second security system **924***b* may include more or fewer security devices, such as, for example, one firewall **905***b* and no IPS. First and second security devices **924***a/b* may be identically configured. First and second security systems **924***a/b* may be connected via a link **929**. Link **929** may be a secure link. Link **929** may be an internal link to LAN **900** or alternatively, a link that is part of an external network. Second security system **924***b* may be directly coupled to an external network, e.g., the Internet, using external network interface **902***b*. Alternatively, second security system **924***b* may be coupled to the external network through first security system **924***a*. Similarly, first and second security systems **924***a/b* may share a single internal network interface. Computers, servers and other devices in the LAN **900** may be interconnected using a number of data transmission media, including, but not limited to wire, fiber optics, and radio waves.

Other configurations for a high availability network topology are possible. In each configuration, second security system **924***b* may act at failover to support traffic processed by first security system **924***a*. In one implementation consistent with the principles of the invention, second security system **924***b* may be provided by a pool of security systems. In the pool, at least one security system may be identified as a primary failover system. One or more other security systems in the pool may be identified as secondary failover systems. Each of the second security systems may be passive (i.e., idle until a failover event) or actively processing packets in support of its own network requirements. In one implementation consistent with the principles of the invention, a first and second security system may each provide failover protection for the other. In such an implementation, failover data may be exchanged between the two security systems. The operation of the security systems prior to, and in support of, failover is discussed in greater detail below.

In a high availability implementation shown in FIG. **9**, second security system **924***b* may be configured to operate at failover of first security system **924***a*. Failover may arise when either first security system **924***a*, or links to first security system **924***a*, fail in the network topology. Failover may be detected by a failure to receive a keep-alive signal, data or other status information by second security system **924***b*. In one implementation consistent with the principles of the invention, first security system **924***a* may include a failover engine **930**. Failover engine **930** may be operable to transmit failover data to another security system (e.g., second security system **924***b*) so as to synchronize the two security systems. In one implementation consistent with the principles of the invention, the failover data may include data from flow table **215** (part of session module **122**) associated with a respective security system (e.g., first security system **924***a*). More specifically, after failover, a second security system (e.g., second security system **924***b*) may receive packets for routing and processing ordinarily (i.e., but for the failure) destined for processing by the first security system (e.g., first security system **924***b*). Some of the packets received relate to sessions that have previously been processed and identified in the first security system. Sharing the flow information prior to failover may allow the second security system to seamlessly process packets for existing flows as they are received. Conventional systems that do not share flow information prior to failover may be required to drop all packets that relate to existing sessions (i.e., sessions that were current at the time of failover) or, alternatively, repeat processing steps.

As described above, session module **122***a* within first security system **924***a* may monitor packets being communicated within the network. Session module **122***a* may act in conjunction with firewall device **905***a* and IPS device **910***a* to facilitate blocking of packets associated with attempted network security intrusions.

A failover engine **930** of secondary security system **924***b* (FIG. **9**) may operate to detect a failure in one or more primary security systems (e.g., the first security system) for which a given security system may be designated to act as a failover device. Failover engine **930** may operate to detect failures in the links or operation of a given primary security system. In one implementation consistent with the principles of the invention, a given security system may act as a failover system for a one or more other security systems. Failover engine **930** may control receipt of and update of information for flow table **215** in a respective device. More specifically, failover engine **930** may be operable to provide synchronization information from a primary security system to a secondary security system, update the synchronization information over time, detect the failure of the primary security system and initiate the processing of packets in the secondary security system for packets that otherwise would have been processed by the primary security system but for the detected failure.

Referring now to FIG. **10**, another implementation of session module **122** is illustrated. This implementation of session module **122** may include incoming packet interface **205** for receiving packets. The received packets may be analyzed by flow processing engine (FPE) **202** to determine if an attempted network security intrusion is in progress. Session module **122** may also includes flow table **215**. Flow table **215** may be used to store information regarding flows associated with received packets. Flow table **215** may include a primary or active portion **1002** and a secondary portion **1004**. Primary portion **1002** may be that portion of the flow table dedicated to store information related to the operation of the given session module as a primary security system (e.g., store flow information for which the session module is actively participating in the processing of the packets). Secondary portion **1004** may be that portion of the flow table dedicated to store information related to the operation of the given session module as a secondary security system (e.g., failover/synchronization information for flows that the session module may process in the event of a failover). In one implementation, the primary and secondary portions **1002** and **1004** may be integrated in flow table **215**. In another implementation, flow table **215** may store multiple secondary portions corresponding to multiple primary security systems for which a given session module may be providing failover support.

Session module **122** may also include interfaces to other security devices on the network as well as one or more interfaces to other security systems. In one implementation consistent with the principles of the invention, session module **122** may include a firewall interface **220**, an IPS interface **225**, and a failover interface **1000**. The security device interfaces are used by session module **122** to obtain information regarding the received packet, and information regarding the flow associated with the packet, in order to determine if the received packet should be allowed or modified. The security device interfaces may also be used by session module **122** to communicate flow information that the security devices may use to facilitate processing of the packet. Failover interface **1000** may be used to transmit synchronization information to other security systems in the network.

In one implementation consistent with the principles of the invention, each of the primary and secondary security systems may include session modules **122** that may include failover engine **930**. In such an implementation, first security system **924***a* and second security system **924***b* (FIG. **9**) may not have a separate failover engine **930** outside of session module **122**.

The processing steps for the first security system (i.e., a primary security system that operates to conventionally pass packets) may include the initialization of a flow table **215** in session module **122** (act **1102**). After initialization, failover engine **930** may identify information reflecting the initial/current state of the flow table **215** (act **1104**) and may pass the information to a second security system through failover interface **1000** (where it is stored, e.g., in secondary portion **1004**) (act **1106**). After initialization, session module **122** for the first security system may analyze packets and develop further session and other information that is stored in or deleted from flow table **215** as described above with respect to FIGS. **3-5** (act **1108**). At predetermined times, failover engine

930 may provide information from flow table **215** to one or more second security systems through failover interface **1000** (act **1110**). This process may repeat so as to maintain a current copy of information from flow table **215** in the second security systems (e.g., in respective secondary portions of flow table **215**).

In one implementation, flow table **215** may be copied and provided in its entirety to the second security system at predetermined times. In an alternative implementation, only portions of flow table **215** may be copied. In one implementation, a message is sent each time a session is created or torn down in the first security system. In one implementation, time-out information may be provided for each new session. In this implementation, a refresh message may be sent to each second security system whenever an associated timer in the first security system is reset (i.e., refreshing the timers in the second security systems). Alternatively, no time-out information may be sent with the session information passed to the second security systems. In this configuration, the second security systems may receive refresh messages at set-up and tear down of sessions in the first security system.

In one implementation, time-out information may be provided for each new session created. In the second security systems, the time-out function may be disabled (i.e., the second security systems will not delete the session from the flow table after the time-out period has expired). In one implementation, only when a second security system takes over packet processing after a failover may the timers be activated for the traffic associated with the first security system.

The processing acts for the second security system (e.g., the security system that operates to process packets from the first security system at failover) may include the initialization of flow table **215** in session module **122** of second security system (act **1122**). Information reflecting the initial state of flow table **215** may be received by the second security system through failover interface **1000** (act **1124**) and stored in secondary portion of flow table **215** (act **1126**). The second security system may continue to receive updates from the first security system at predetermined intervals (act **1128**) and flow table **215** may be updated (act **1126**). When a failover is detected by failover engine **930** (act **1130**), session module **122** for the secondary security system may initialize flow table **215** (act **1132**). Failover may be detected by an external entity or by the second security system. Packets may be provided to both the first and the second security system for processing. However, the second security system may be configured to not process packets unless a failover has been detected. Failover detection may be detected by ping or keep-alive signals. In one implementation, the first security system may provide a keep-alive signal to the second security system. Alternatively, the second security system may ping the first security system intermittently to determine whether the first security system is operational. In another implementation, external entities may monitor the operation of the first security system. Upon detection of a fault in either the first security system or the connection paths associated therewith, a take-over signal can be generated and passed to an appropriate second security system.

Initializing flow table **215** in the second security system may include activating an appropriate secondary portion **1004** of flow table **215**. Initialization may include the reordering of flow table **215** to integrate records of the primary and secondary portions **1002** and **1004** respectively (e.g., if the second security system is actively supporting other packet processing prior to failover). In one implementation, each record may include a label indicating to which security system the record belongs. The label may be used to easily clear records from the primary portion of the flow table of the second security system in the event the primary security system is recovered. Thereafter, session module **122** in the second system may begin to receive and analyze packets (developing further session and other information that is stored in/deleted from flow table **215**) as described above with respect to FIGS. **3-5** (act **1134**). At predetermined times, failover engine **930** of session module **122** may provide information from flow table **215** to one or more third security systems through its failover interface **1000** (act **1136**). Update information may be provided at predetermined intervals to the third security system(s) so as to maintain a current copy of information from flow table **215** in the one or more third security systems. In one implementation, flow table **215** may be copied and provided in its entirety to the one or more third security systems at predetermined times. In an alternative implementation, only portions of flow table **215** may be copied. In one implementation, the process may continue to operate on packets until a predetermined event occurs. The predetermined event may be the recovery from the failure in the first security system.

Referring now to FIG. **11***c*, a process for providing high availability in a security network is shown. A primary security system may be identified (act **1152**) and one or more secondary security systems may be identified (act **1154**). The primary security system may be a security system that operates to process packets that the primary security system receives until a failover event occurs. Failover may include failure of the primary security system as well as links from the primary security system to the network. The secondary security systems may be designated as available for processing the load of packets associated with the primary security system after a failover event. In one implementation, one secondary security system may be identified. In an alternative implementation, a pool of secondary security systems may be identified. In one implementation of the pool, one secondary system may be designated as a master and one or more other secondary security systems may be designated as slaves. At failover, the master security system may act to process packets of the failed primary security system. The slaves may operate to take the place of the master upon failure of the master device.

Returning to the high availability process, the primary security system may initialize flow table **215** (act **1156**) and the initial configuration of the flow table may be passed to the secondary security system(s) (act **1158**). Packets may be conventionally processed by the primary security system and flow table **215** may be updated accordingly (act **1160**). At predetermined times, flow table **215** from the secondary system may be updated with information from the primary security system (act **1162**). When a failover event is detected (act **1164**), the secondary security system may initialize flow table **215** in the secondary system and may begin to process packets routed to the primary security system (act **1166**). In one implementation, upon correction of the failover event, the primary security system may be reinitialized, including updating flow table **215**, and packet processing may be resumed by the primary security system.

Where as described above with respect to FIG. **9**, the primary and secondary security systems include multiple security devices (e.g., a firewall and IPS), no separate synchronization of the individual devices may be required. Accordingly, the amount of data to be passed between the respective devices is minimized. Further, reliability may be increased with the minimization of information that is to be passed. In the configuration shown in FIG. **9**, each of the security devices may share information in unified flow table **215**. Other processing efficiencies can be realized in the secondary security system. For example, at failover, packets that were previously identified as being part of a recognized flow may bypass processing by the IPS. In one implementation, two security systems may act as failover devices for each other (e.g., first system acts as failover for the second system and visa versa). In other implementations, the failover security system may be a conventional system (i.e., no shared flow information among devices) and the flow information that is received by the failover security system may be shared with multiple devices.

Embodiments consistent with the principles of the invention may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Embodiments may be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program may be written in any form of programming language, including compiled or interpreted languages, and it may be deployed in any form, including as a standalone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program may be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Method acts of the invention may be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method acts may also be performed by, and apparatuses may be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor may receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer may also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data may include all forms of nonvolatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory may be supplemented by, or incorporated in special purpose logic circuitry.

Embodiments consistent with the principles of the invention may be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user may interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system may be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include, for example, a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system may include clients and servers. A client and server may be remote from each other and may interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

This invention has been described in terms of particular embodiments. Nevertheless, it will be understood that various modifications may be made without departing with the spirit and scope of the invention. For instance, the steps of the invention may be performed in a different order and still achieve desirable results. In addition, the session module, IPS, firewall, and router may all be incorporated into a single device such as the configuration shown in FIG. **9**. Other configurations of a session module packaged with one or more security devices are also possible. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A method in a computer network, comprising:

processing packets, by a primary security system, the primary security system including a first device-implemented session module to maintain flow information for the primary security system to facilitate processing of the packets, where the first device-implemented session module includes a first flow table having a primary portion that stores information associated with the operation of the first device-implemented session module, when the primary security system is functioning in a primary security system mode, and a secondary portion that stores information associated with the operation of the first device-implemented session module, when the primary security system is functioning in a failover mode;

designating a secondary security system for processing packets upon a failover event, the secondary security system including a second device-implemented session module to maintain flow information for the secondary security system to facilitate processing of the packets, where the second device-implemented session module includes a second flow table having a primary portion that stores information associated with the operation of the second device-implemented session module, when the secondary security system is functioning in a primary security system mode, and a secondary portion that stores information associated with the operation of the second device-implemented session module, when the secondary security system is functioning in a failover mode;

sharing flow records from the primary security system with the secondary security system;

sharing flow records from the secondary security system with the primary security system;

using the primary security system to provide failover support for the secondary security system, based on the information stored in the secondary portion of the first flow table; and

using the secondary security system to provide failover support for the primary security system, based on the information stored in the secondary portion of the second flow table.

2. The method of claim **1**, further comprising:

determining whether the failover event occurred; and

processing a packet by one of the primary security system or the secondary security system for the other one of the primary security system or the secondary security system when the failover event occurs.

3. The method of claim **1**, where the failover event includes the failure of one of the primary security system or the secondary security system.

4. The method of claim **1**, where the failover event includes a failure of a link from one of the primary security system or the secondary security system to the computer network.

5. The method of claim **2**, where the determining is performed at the one of the primary security system or the secondary security system.

6. The method of claim **2**, where the determining further comprises:

detecting an absence of a keep-alive signal.

7. The method of claim **2**, where the determining further comprises:

monitoring operation of the one of the primary security system or the secondary security system, and

sending a take-over signal to the other one of the primary security system or the secondary security system when the monitoring operation detects a fault.

8. The method of claim **1**, where the secondary security system is configured substantially identical to the primary security system.

9. The method of claim **1**, where the sharing flow records further comprises:

sharing the flow records between the primary security system and the secondary security system at predetermined intervals.

10. The method of claim **1**, where the sharing flow records further comprises:

sharing the flow records between the primary security system and the secondary security system when a refresh message is received by one of the primary security system or the secondary security system.

11. The method of claim **10**, where the one of the primary security system or the secondary security system sends the refresh message when a session is set-up or torn down.

12. The method of claim **1**, further comprising:

resuming receiving and processing of a packet at one of the primary security system or the secondary security system when a condition that caused the failover event is cleared.

13. A system, comprising:

a processor-implemented primary security system to process packets, the primary security system including a first device-implemented session module to maintain flow information for the primary security system to facilitate processing of the packets, where the first device-implemented session module includes a first flow table having a primary portion that stores information associated with an operation of the first device-implemented session module, when the primary security system is functioning in a primary security system mode, and a secondary portion that stores information associated with an operation of the first device-implemented session module, when the primary security system is functioning in a failover mode; and

a secondary security system to process packets upon a failover event, the secondary security system including a second device-implemented session module to maintain flow information for the secondary security system to

facilitate processing of packets, where the second device-implemented session module includes a second flow table having a primary portion that stores information associated with an operation of the second device-implemented session module, when the secondary security system is functioning in a primary security system mode, and a secondary portion that stores information associated with an operation of the second device-implemented session module, when the secondary security system is functioning in a failover mode,

where the primary security system and the secondary security system share flow records, and

where the primary security system is to provide failover support for the secondary security system, based on the information stored in the secondary portion of the first flow table and the secondary security system is to provide failover support for the primary security system, based on the information stored in the secondary portion of the second flow table.

14. The system of claim **13**, where

one of the primary security system or the secondary security system is to determine if the failover event has occurred, and

when the failover event is determined to have occurred, one of the primary security system or the secondary security system is to process the packets for the other one of the primary security system or the secondary security system.

15. The system of claim **13**, where the failover event includes failure of one of the primary security system or the secondary security system.

16. The system of claim **13**, where the failover event includes a failure of a link from one of the primary security system or the secondary security system to a computer network.

17. The system of claim **14**, where the one of the primary security system or the secondary security system is to determine if the failover event has occurred by detecting an absence of a keep-alive signal.

18. The system of claim **14**, where the one of the primary security system or the secondary security system is to determine if the failover event has occurred by detecting a fault in the one of the primary security system or the secondary security system.

19. The method of claim **13**, where the secondary security system is substantially identical to the primary security system.

20. The system of claim **13**, where the primary security system and the secondary security share the flow records at predetermined intervals.

21. The system of claim **13**, where the primary security system and the secondary security share the flow records when a refresh message is received by one of the primary security system or the secondary security system.

22. The system of claim **21**, where the one of the primary security system or the secondary security system sends the refresh message when a session is set-up or torn down.

23. The system of claim **13**, where one of the primary security system or the secondary security system resumes receiving and processing of packets when a condition that caused the failover event is cleared.

* * * * *

EXHIBIT G

US007107612B1

(12) **United States Patent**
Xie et al.

(10) **Patent No.: US 7,107,612 B1**
(45) **Date of Patent: Sep. 12, 2006**

(54) **METHOD, APPARATUS AND COMPUTER PROGRAM PRODUCT FOR A NETWORK FIREWALL**

(75) Inventors: **Ken Xie**, Atherton, CA (US); **Yan Ke**, San Jose, CA (US); **Yuming Mao**, Milpitas, CA (US)

(73) Assignee: **Juniper Networks, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 14 days.

(21) Appl. No.: **10/893,283**

(22) Filed: **Jul. 19, 2004**

Related U.S. Application Data

(63) Continuation of application No. 09/525,369, filed on Mar. 15, 2000, now Pat. No. 6,772,347, which is a continuation-in-part of application No. 09/283,730, filed on Apr. 1, 1999, now Pat. No. 6,701,432.

(51) **Int. Cl.**
G06F 7/04 (2006.01)
G06F 9/00 (2006.01)

(52) **U.S. Cl.** **726/13**; 726/14; 726/6; 726/7

(58) **Field of Classification Search** 726/14, 726/13, 12, 11, 7, 21, 26, 2–6
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,606,668	A	2/1997	Shwed	395/200.11
5,835,726	A	11/1998	Shwed et al.	395/200.59
5,951,651	A	9/1999	Lakshman et al.	709/239
5,983,270	A *	11/1999	Abraham et al.	709/224
6,009,475	A	12/1999	Shrader	709/249
6,016,310	A	1/2000	Muller et al.	370/255
6,400,707	B1	6/2002	Baum et al.	370/352
6,701,432	B1	3/2004	Deng et al.	713/153
6,757,680	B1 *	6/2004	Choy	707/9
6,772,347	B1 *	8/2004	Xie et al.	726/11
6,845,452	B1 *	1/2005	Roddy et al.	726/11
7,013,482	B1 *	3/2006	Krumel	726/13
2002/0027907	A1 *	3/2002	Tateoka	370/389
2002/0188720	A1	12/2002	Terrell et al.	709/225

FOREIGN PATENT DOCUMENTS

EP	0 658 837 A1	6/1995
EP	0 893 921 A1	1/1999

OTHER PUBLICATIONS

Internet Security Systmes, http://www.iss.net/security_center/advice/intrusions/2001323, pp. 1-2.*

Tsuchiya, Paul F., "Extending the IP Internet through address reuse," Jan. 1993, ACM Sigcomm, vol. 23, Issue No. 1., pp. 16-33.*

Lodin, Steven et al., "Firewalls fend off invasion from the net," IEEE Spectrum Feb. 1998, pp. 26-34.*

"Network Address Translation Technical Discussion," 1996, http://www.safety.net/nattech.html, pp. 1-4.*

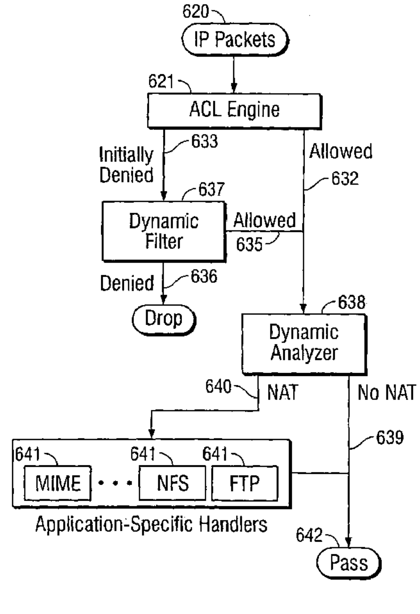
* cited by examiner

Primary Examiner—Norman M. Wright
(74) *Attorney, Agent, or Firm*—Harrity Snyder, LLP

(57) **ABSTRACT**

An improved firewall for providing network security is described. The improved firewall provides for dynamic rule generation, as well using conventional fixed rules. This improvement is provided without significant increase in the processing time required for most packets. Additionally, the improved firewall provides for translation of IP addresses between the firewall and the internal network.

28 Claims, 6 Drawing Sheets



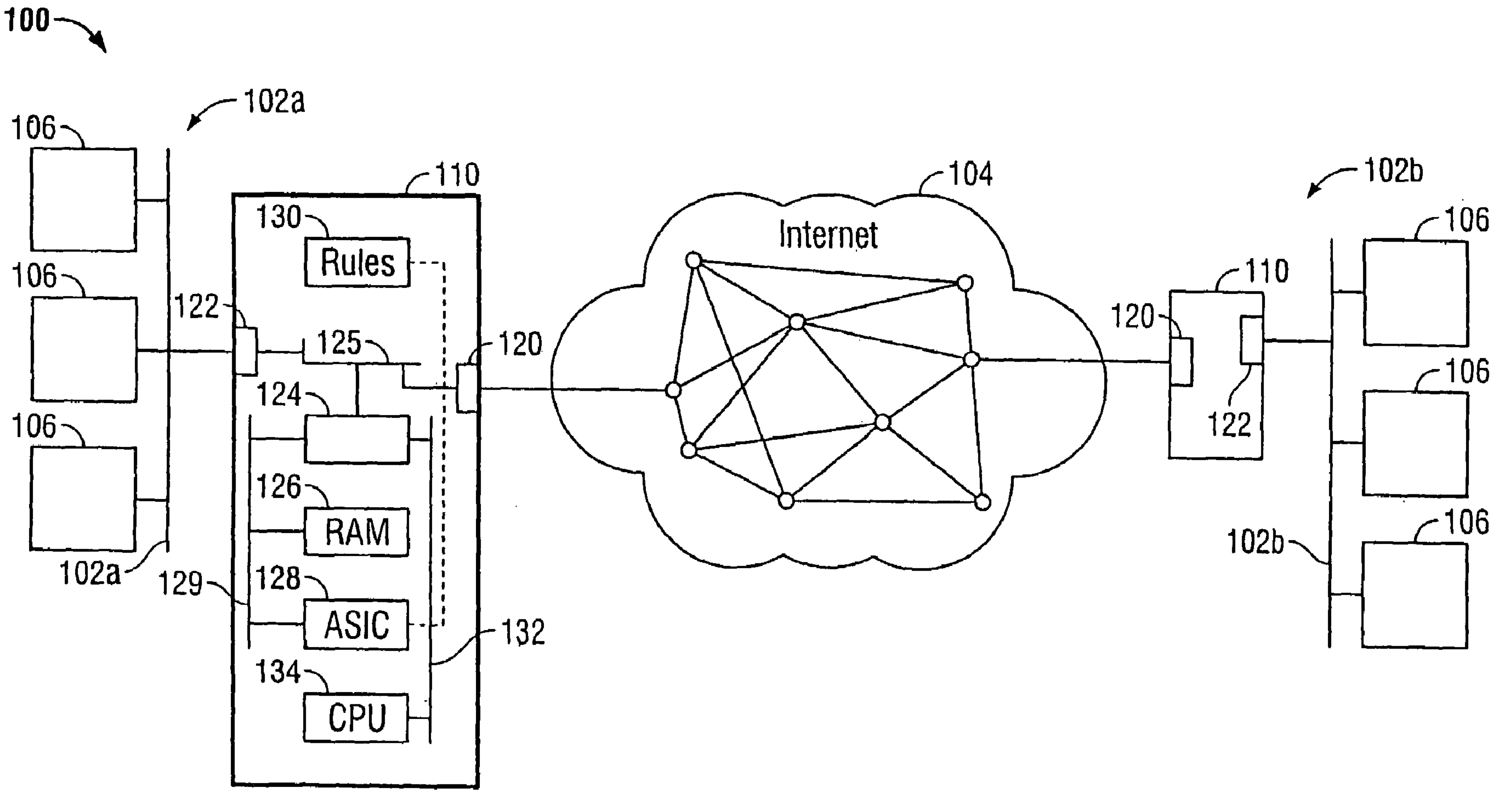


FIG. 1 (Prior Art)

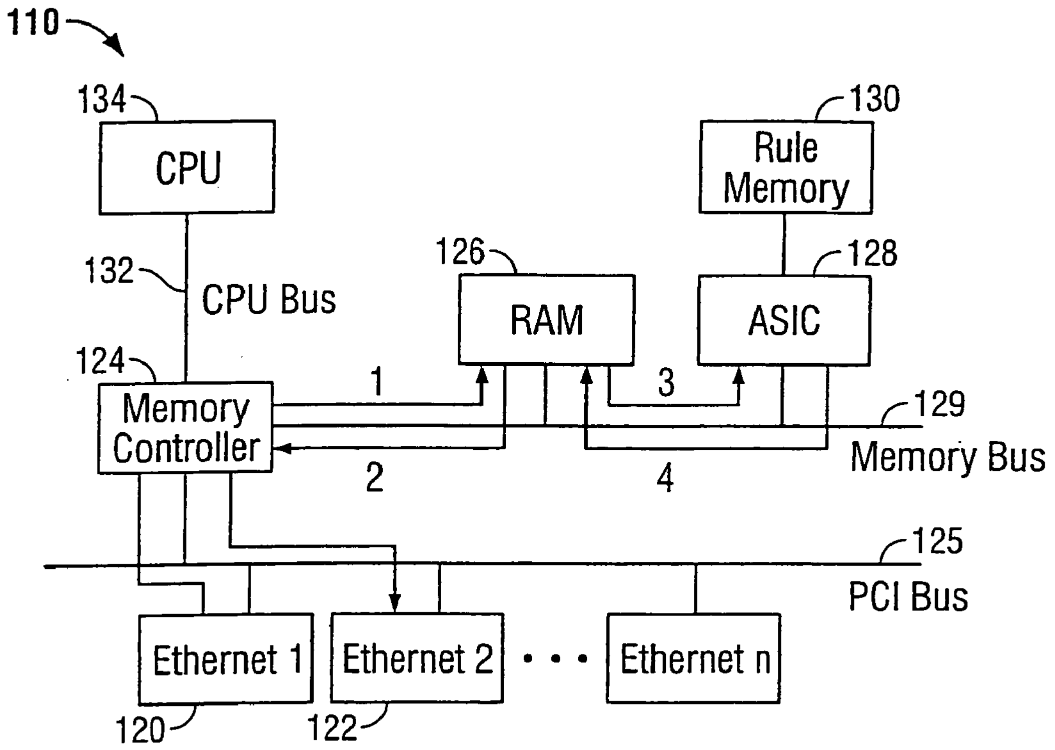


FIG. 2A (Prior Art)

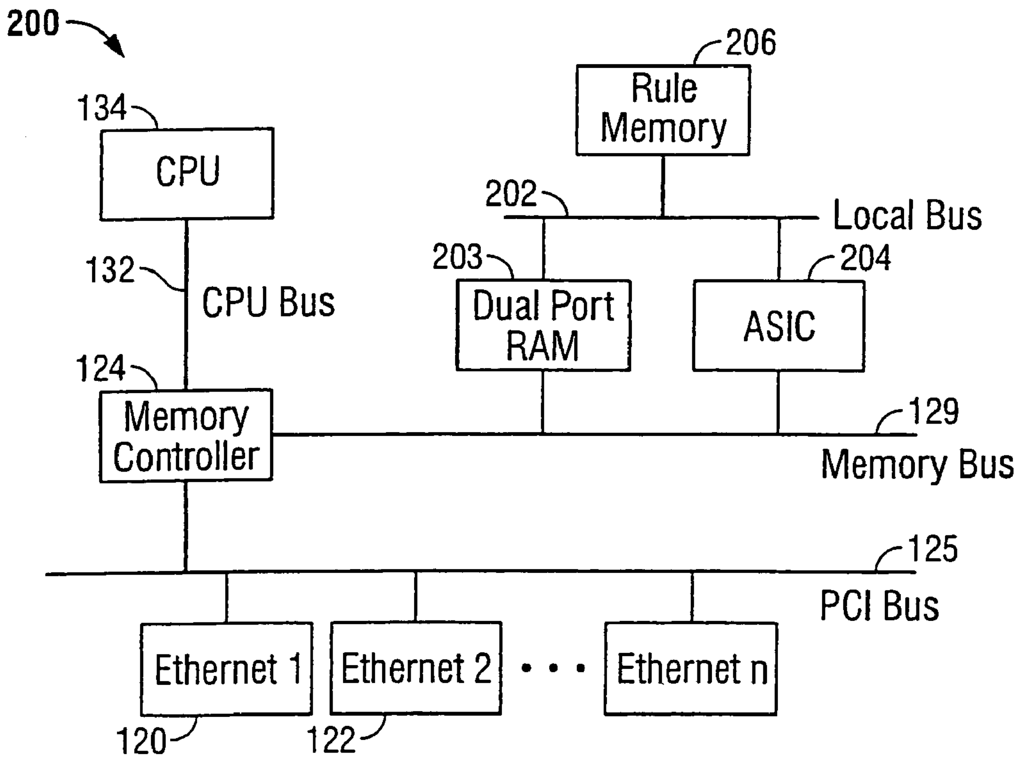


FIG. 2B

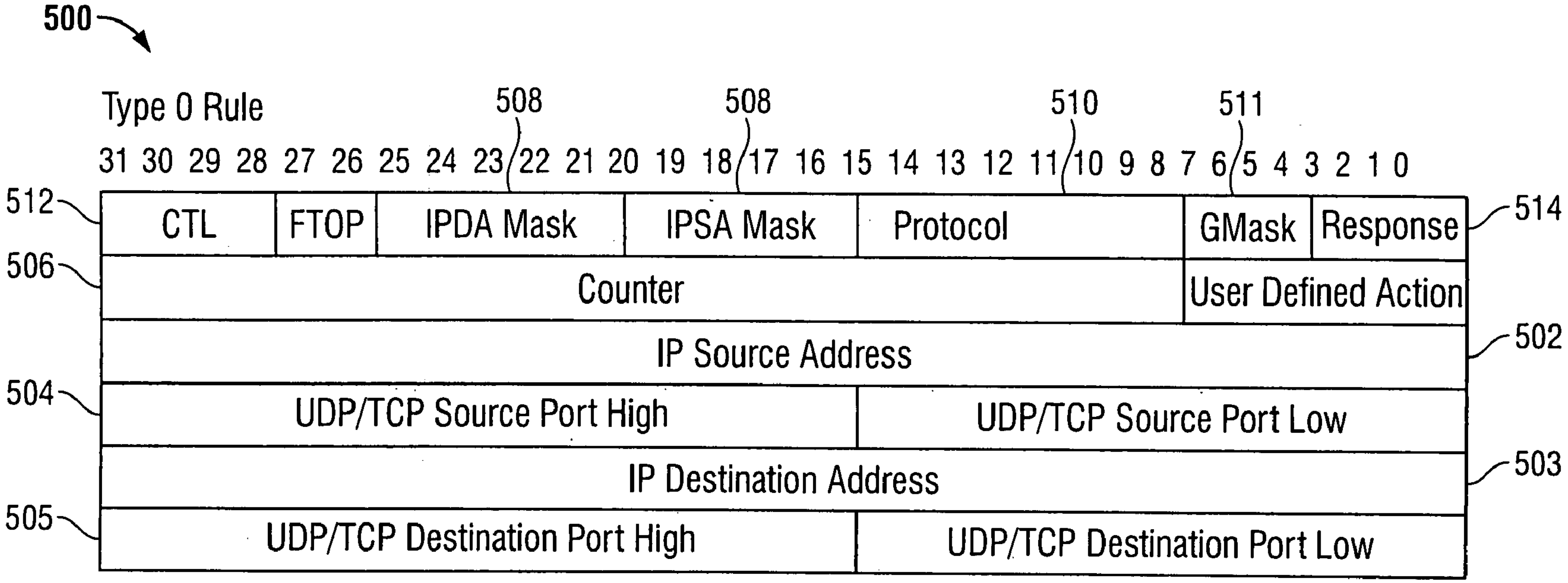

500
Type 0 Rule
508
508
510
511
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
512
CTL
FTOP
IPDA Mask
IPSA Mask
Protocol
GMask
Response
514
506
Counter
User Defined Action
IP Source Address
502
504
UDP/TCP Source Port High
UDP/TCP Source Port Low
IP Destination Address
503
505
UDP/TCP Destination Port High
UDP/TCP Destination Port Low

FIG. 3

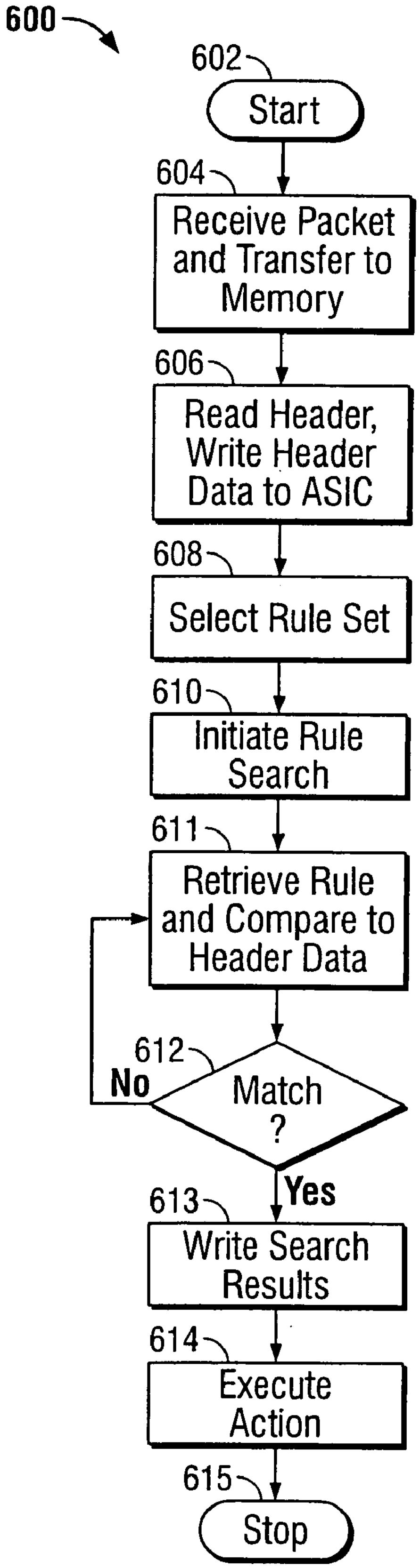

600
602
Start
604
Receive Packet and Transfer to Memory
606
Read Header, Write Header Data to ASIC
608
Select Rule Set
610
Initiate Rule Search
611
Retrieve Rule and Compare to Header Data
612
No
Match ?
Yes
613
Write Search Results
614
Execute Action
615
Stop

FIG. 4

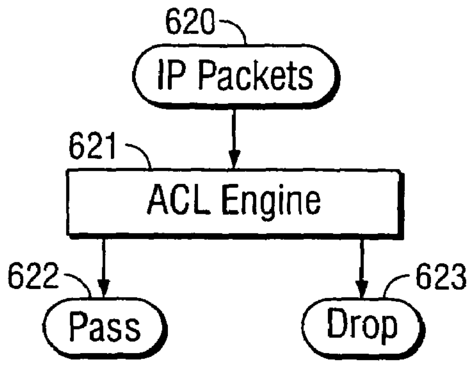


FIG. 5
(Prior Art)

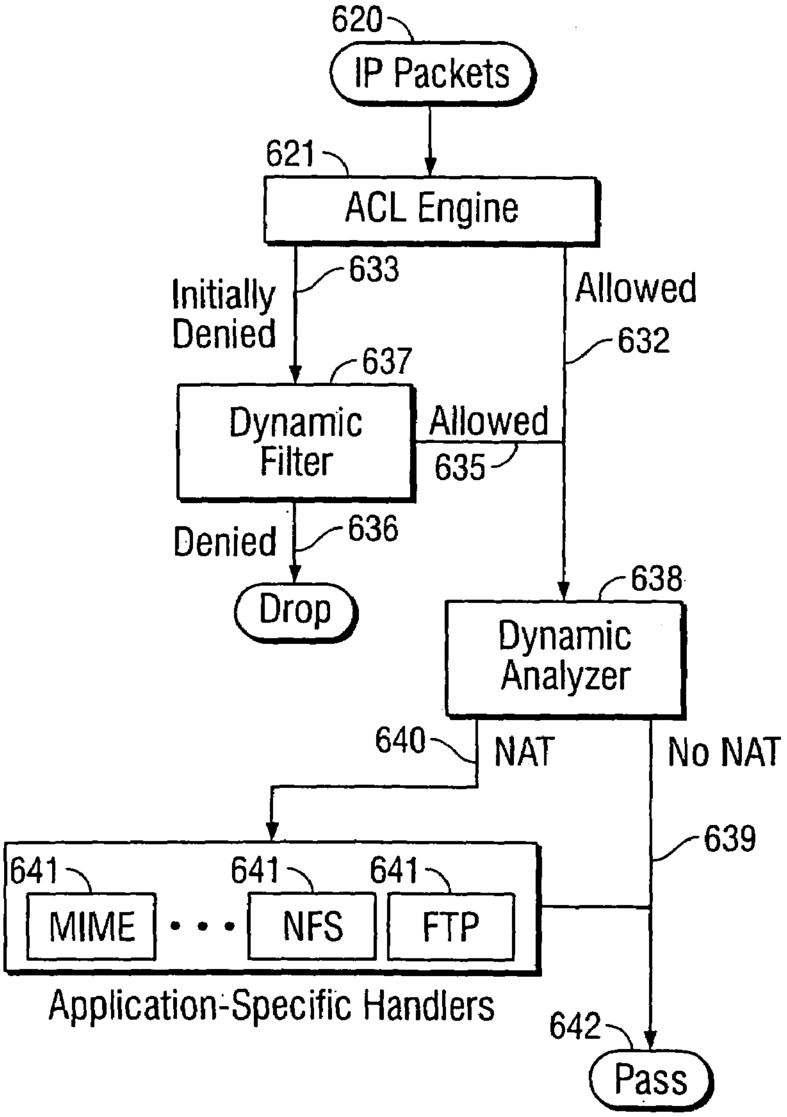


FIG. 6

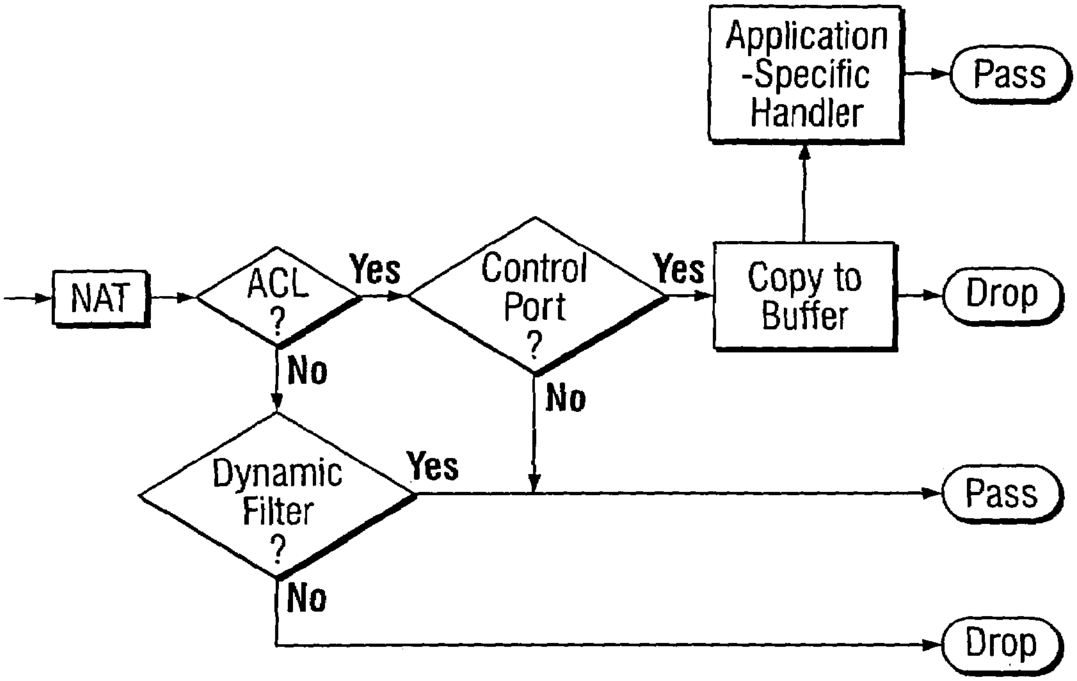

NAT
ACL ?
Yes
No
Control Port ?
Yes
No
Copy to Buffer
Application -Specific Handler
Pass
Drop
Dynamic Filter ?
Yes
No
Pass
Drop

FIG. 7A

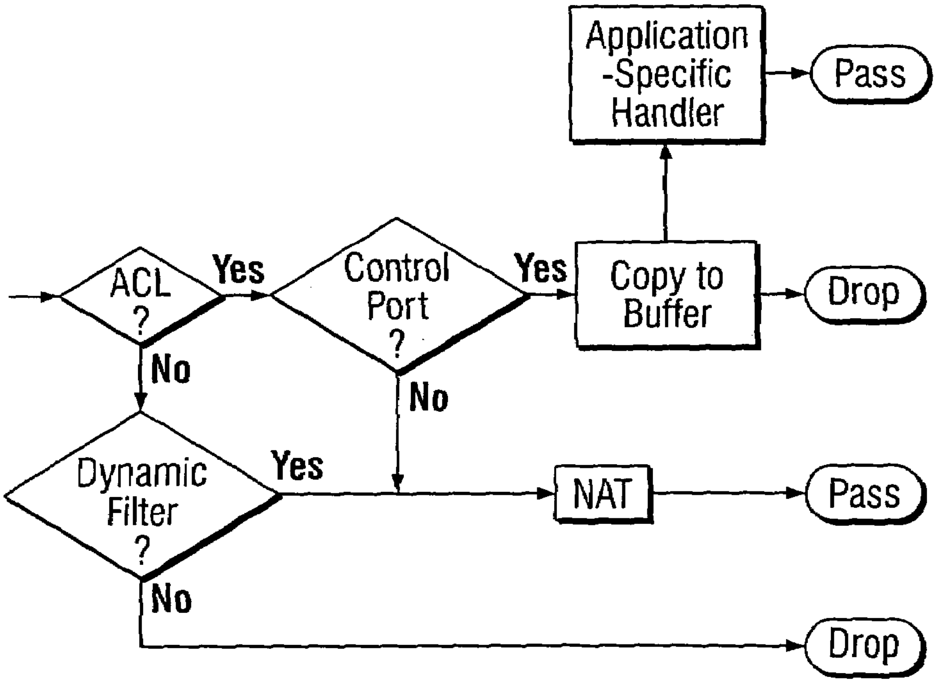

ACL ?
Yes
No
Control Port ?
Yes
No
Copy to Buffer
Application -Specific Handler
Pass
Drop
Dynamic Filter ?
Yes
No
NAT
Pass
Drop

FIG. 7B

METHOD, APPARATUS AND COMPUTER PROGRAM PRODUCT FOR A NETWORK FIREWALL

RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 09/525,369 filed Mar. 15, 2000, which is now U.S. Pat. No. 6,772,347 continuation-in-part of co-pending application Ser. No. 09/283,730, filed on Apr. 1, 1999, now U.S. Pat. No. 6,701,432 B1, issued on Mar. 2, 2004, the disclosures of which are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to the field of computer networks. In particular, the present invention relates to a method, apparatus and computer program product for providing network security.

BACKGROUND OF THE INVENTION

A packet switch communication system includes a network of one or more routers connecting a plurality of users. A packet is the fundamental unit of transfer in the packet switch communication system. A user can be an individual user terminal or another network. A router is a switching device that receives packets containing data or control information on one port and, based on destination information contained within the packets, routes the packets out another port to their final destination, or to some intermediary destination(s). Conventional routers perform this switching function by evaluating header information contained within the packet in order to determine the proper output port for a particular packet.

As known, a communications network can be a public network, such as the Internet, in which data packets are passed between users over untrusted, i.e., non-secure communication links. Alternatively, various organizations, typically corporations, use what is known as an intranet communications network, accessible only by the organization's members, employees, or others having access authorization. Intranets typically connect one or more private servers, such as a local area network (LAN). The network configuration in a preferred embodiment of this invention can include a combination of public and private networks. For example, two or more LANs can be coupled together with individual terminals using a public network, such as the Internet. A network point that acts as an entrance to another network is known in the art as a gateway.

Conventional packet switched communication systems that include links between public and private networks typically include means to safeguard the private networks against intrusions through the gateway provided at the interface of the private and public networks. The means designed to prevent unauthorized access to or from a private are commonly known as firewalls, which can be implemented in both hardware and software, or a combination of both. Thus, a firewall is a device that can be coupled in-line between a public network and a private network for screening packets received from the public network.

Referring to FIG. **1**, a conventional packet switch communication system **100** can include two (or more) private networks **102***a* and **102***b* coupled by a public network **104** for facilitating the communication between a plurality of user terminals **106**. Each private network **102** can include one or more servers and a plurality of individual terminals.

Each private network **102** can be an intranet, such as a LAN. Public network **104** can be the Internet, or other public network having untrusted links for linking packets between private networks **102***a* and **102***b*. In a preferred embodiment, at each gateway between a private network **102** and public network **104** there is a firewall **110**.

The architecture of an illustrative prior art firewall is shown in FIG. **2***a*. The firewall **110** generally includes one or more public network links **120**, one or more private network links **122**, and memory controller **124** coupled to the network links by a PCI bus **125**. Memory controller **124** is also coupled by a memory bus **129** to a memory (RAM) **126** and a firewall engine, implemented in a preferred embodiment as an ASIC **128**. The firewall engine ASIC **128** performs packet screening prior to routing packets through to private network **102**. The firewall engine ASIC **128** processes the packets to enforce an access control policy, screening the packets in accordance with one or more sets of rules. The rules are described in more detail below. A central processor (CPU) **134** is coupled to memory controller **124** by a CPU bus **132**. CPU **134** oversees the memory transfer operations on all buses shown. Memory controller **124** is a bridge connecting CPU bus **132**, memory bus **129**, and PCI bus **125**.

In operation, packets are received at public network link **120**. Each packet is transferred on bus **125** to, and routed through, memory controller **124** and on to RAM **126** via memory bus **129**. When firewall engine **128** is available, packets are fetched using memory bus **129** and processed by the firewall engine **128**. After processing, the packet is returned to RAM **126** using memory bus **129**. Finally the packet is retrieved by the memory controller **124** using memory bus **129**, and routed to private network link **122**. The screening rules implemented by the firewall engine **128** are typically searched in linear order, beginning with the internal rule memory. Certain aspects of the rule structure are described below.

As known in the art, a rule is a control policy for filtering incoming and outgoing packets. Rules specify actions to be applied as against certain packets. When a packet is received for processing through a rule search, the packet's IP header, TCP header, or UDP header may require inspecting. A rule will generally include, at a minimum, source/destination IP addresses, UDP/TCP source/destination ports and transport layer protocol. Additional criteria may be used by the rules as well.

Generally, the address information is used as matching criterion—in other words to match a rule, a packet must have come from a defined source IP address and its destination must be the defined destination IP address. The UDP/TCP source/destination port specifies what client or server process the packet originates from on the source machine. The firewall engine can be configured to permit or deny a packet based upon these port numbers. The rule may include a range of values or a specific value for a TCP/UDP port. The transport layer protocol specifies which protocol above the IP layer, such as TCP or UDP, the policy rule is to be enforced against.

The firewall engine described above essentially screens packets using an access control list (ACL), and may be referred to as an ACL engine. That is, it performs a simple comparison of various matching criteria of an incoming IP packet—typically source, destination, port and protocol—to each rule in a rule set in sequence. Based upon this comparison, an incoming IP packet is either allowed or denied. A data-flow chart for this firewall engine is shown in FIG. **5**.

It will be appreciated that using a fixed set of rules can be restrictive in many practical applications. Therefore, it is desirable to provide a system and method capable of adding rules to the rule set of the firewall engine dynamically—that is, to extract from a sequence of packets information, such as the port number and IP address, and generate new rules using this information. However, generating these new rules dynamically would increase the complexity of the comparison and decrease the speed of the firewall engine. There is therefore a need in the art for a firewall engine which can generate rules dynamically, based upon information extracted from incoming packets, with a limited impact on the speed of the firewall engine.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment, an apparatus, method and computer program product for providing network security is described. The apparatus includes an engine for sorting incoming IP packets into initially allowed and initially denied packets using a fixed set of rules. The packets are then further sorted by a second engine. In one embodiment, the engine further sorts the initially denied packets into allowed packets and denied packets, using dynamically-generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

Likewise, the method includes the step of sorting incoming IP packets into initially allowed and initially denied packets using a fixed set of rules. The packets are then further sorted. In one embodiment, additional steps include sorting the initially denied packets into allowed packets and denied packets, using dynamically generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

Finally, the computer program product sorts incoming IP packets into initially allowed packets and initially denied packets. In one embodiment, the computer program product further sorts the initially denied packets into allowed packets and denied packets, using dynamically generated rules. The denied packets are dropped and the allowed packets are permitted to enter the network.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates an exemplary packet switch communications system.

FIG. **2***a* illustrates a firewall with an application-specific integrated circuit (ASIC).

FIG. **2***b* illustrates a firewall with a local bus and an application-specific integrated circuit (ASIC).

FIG. **3** illustrates an exemplary rule structure for use by a firewall.

FIG. **4** is a flow diagram for a firewall screening process.

FIG. **5** is a data-flow chart for a prior art firewall.

FIG. **6** is a data-flow chart for a firewall in accordance with one embodiment of the invention.

FIG. **7***a* is a logic diagram for processing incoming packets in accordance with the invention.

FIG. **7***b* is a logic diagram for processing outgoing packets in accordance with the invention.

DETAILED DESCRIPTION OF THE INVENTION

A conventional firewall may be implemented in software, or in hardware as shown in FIG. **2***a*. Alternatively, a hybrid

of software and hardware may also be used to implement a firewall. The firewall of FIG. **2***a* uses a memory bus **129** to communicate between the ASIC **128**, the RAM **126**, and the memory **130**, which stores the rules used by the firewall. FIG. **2***b* shows a high-speed firewall that employs a local bus **202** for improved processing speed. A high-speed firewall is described in pending parent application Ser. No. 09/283,730, the contents of which is hereby incorporated by reference. Exemplary high-speed firewalls include NetScreen Technology, Inc.'s integrated firewall products, described at www-.netscreen.com and related web pages. Selected web pages describing NetScreen's high-speed firewalls are provided as Appendix A to this application.

As shown in FIG. **2***b*, the high-speed firewall includes a hardware ASIC **204** to implement the firewall engine. The firewall engine retrieves packets stored in memory and processes each packet to enforce an access control policy. The processing by the firewall engine includes retrieving rules from a rule set, and screening the packets in accordance with the retrieved rules. In a specific embodiment, the rules may be stored in an internal memory in the ASIC **204**, or may be retrieved from a separate rule memory **206** via the local bus **202**. In a preferred embodiment, frequently accessed rule sets may be stored in the internal memory, with less-frequently accessed rule sets being stored in the separate rule memory **206**.

The structure **500** of a rule used by a firewall engine in accordance with one embodiment of the present invention is shown in FIG. **3**. A rule will generally include, at a minimum, source/destination IP addresses **502 503**, UDP/TCP source/destination ports **504 505** and transport layer protocol **510**. Additional information used by the rules may include: a range of values for the UDP/TCP source/destination port **504 505**; a counter **506** to keep track of the number of times the rule has been matched; a general mask (GMASK) **511** to indicate whether to ignore or check certain information in the packet header; source/destination IP address mask **508** to indicate whether to ignore part of an IP address, typically a specified number of the least significant bits; a searching control field **512** to tell the firewall engine to search in the separate rule memory **206** and to give a starting address; and a response action field **514** to specify the action to be taken if the rule is matched.

The address information is used as matching criterion—to match a rule, a packet must have come from the defined source IP address **502** and its destination must be the defined destination IP address **503**. Part of the address may be masked using the source/destination IP address mask **508**. The UDP/TCP source/destination port **504 505** specifies what client or server process the packet originates from on the source machine. The firewall engine can be configured to permit or deny a packet based upon these port numbers. The rule may include a range of values or a specific value for a TCP/UDP port. The counter **506** is used to track the number of times a rule has been matched, and at some threshold value will trigger a certain action, such as deny, log or alarm. The transport layer protocol **510** specifies which protocol above the IP layer, such as TCP or UDP, the policy rule is to be enforced against.

Referring to FIGS. **2***b* and **4**, a process **600** executed by the firewall engine in the ASIC **204** is shown for screening packets using both the on-chip and off-chip rule memories. The firewall engine process begins at step **602**. A packet is received at an interface (public network interface **122**) and transferred to dual-ported memory **203** using a DMA process executed by memory controller **124** (**604**).

CPU **134** reads the packet header information from packet memory and writes the packet information into special registers on ASIC **204** (**606**). These registers are mapped onto the system memory space, so CPU **134** has direct access to them. In an exemplary hardware firewall, the registers include: a source IP register; a destination IP register; a port register; a protocol register; and an acknowledge register, for storing the acknowledge bit from the packet.

CPU **134** also specifies which rule set to search by writing to a rule set specifier register (**608**). CPU **134** issues a command to the firewall engine located in the ASIC **204** by writing to a control register to initiate the ASIC rule search (**610**). Alternatively, the firewall engine may first check a stored look-up table with criteria relating to ongoing current applications or services, before searching the rules. In that case, the firewall engine first compares the contents of the special registers to the contents of a look-up table, where the look-up table includes the IP address, port and protocol corresponding to each current application or service. For example, if the packet is an FTP packet for an FTP that is ongoing, this information will be in the look-up table. If, on the other hand, the packet is an FTP packet for a newly-initiated FTP, the information will not be in the look-up table.

If the information is not in the look-up table, or if a look-up table is not used, the firewall engine then compares the contents of the special registers to each rule in sequence (**611**) until a match is found (**612**). The search stops when a match is found (**613**). Alternatively, for certain rules, known as counter rules, the firewall engine will increment the count register and continue the search. If the count threshold is exceeded, or if the search locates a match for a non-counter rule, the search results are written to a status register **616**. Likewise, if no match is found, and the entire set of rules has been examined, the search results are written to the status register. In addition, when a match is found, if a look-up table is used the information identifying the current application, such as the IP address, port and protocol, are written to the look-up table so that later packets in the current application may be processed using the look-up table instead of a rule search.

During the search, CPU **134** polls the status register to check whether the firewall engine is busy or has completed the search. When the CPU **134** determines that the search is complete, the CPU **134** executes certain actions against the current packet based on the information in the status register, such as permit or deny the packet, signal an alarm, and log the packet.

The process described above is a prior art one-pass search process, as illustrated in FIG. **5**: the ACL engine **621** conducts a search through an optional look-up table, and then through rules, as illustrated in FIG. **4**, to determine whether a given packet matches a rule in the set and take action on that basis. The rules use a set of matching criteria—for example, source and destination IP address, and port number, indicating the application. These rules are fixed and use known matching criteria. The ACL engine **621** then allows some packets **622**, and denies or drops, others **623**.

As shown in FIG. **6**, in a preferred embodiment, the IP packets **620** enter the ACL engine **621**. As in the prior art, the ACL engine **621** conducts a search, using fixed rules. The ACL engine then outputs allowed packets **632**, and initially denied packets **633**.

Unlike the prior art, the firewall engine that embodies one aspect of the present invention includes additional dynamic filtering, which further processes the packets. In particular, the initially denied packets **633** are processed by a dynamic filter **637**, which allows some of the initially denied packets to pass through the firewall and enter the private network. The dynamic filter **637** conducts a search through an additional set of rules, which are dynamically generated. The dynamic filter **637** generates rules using criteria such as port number and IP address, which are extracted from incoming packets for applications, such as RealAudio, Netmeeting (which uses the H3232 protocol) and network file system (NFS).

For example, when an FTP is initiated, the first sequence of FTP packets, which includes information on the port number and the IP address, will be passed by the rules in the ACL engine **621**. The dynamic filter **637** then extracts port number and IP address from this first sequence of packets, and generates new rules, similar to the fixed rules used by the ACL, including these criteria. Later sequences of FTP packets will be denied by the ACL engine **621**, but the dynamic filter **637** will pass the packets based on the new, dynamically-generated rules. The way in which the search through the dynamically-generated rules is conducted is similar to the approach used in the ACL engine **621**. The dynamic filter then drops packets which are finally denied **636**, and allows other initially denied packets, which meet the additional access control requirements, to pass **635** through the firewall and enter the private network.

This approach to processing the incoming IP packets has advantages over the prior art. Using dynamically-generated rules allows for more flexible access policy. However, if dynamic rule generation was included in the ACL engine **621**, the processing speed would be decreased. The dynamic filter **637** used in accordance with the present invention, following the ACL engine **621**, advantageously allows the use of dynamically-generated rules, without increasing the processing time for those IP packets, which are initially allowed **632** by the ACL engine **621** based on the fixed rule set.

Another preferred embodiment, as shown in FIG. **6**, additionally allows for network address translation (NAT), to enable IP addresses, port numbers and message authentication codes (MACs) in the private network to be concealed from the public network. The public network can only access this information for the firewall. Thus, the destination information in the headers in the incoming packets must be changed, to reflect the private network IP addresses, port numbers and MAC. Furthermore, source information in the headers of outgoing packets must also be changed, to reflect the firewall network IP address, port number and MAC.

However, depending on the particular application used, information relating to the IP address or port number may be embedded in the packet content or payload, as well as in the header. In that case, the packet payload for an incoming packet must be translated to reflect the internal IP address and port number, as shown in FIG. 7*a*. Likewise, the packet payload for an outgoing packet must be translated to reflect the firewall address and port number, as shown in FIG. 7*b*.

As shown in FIG. **6**, the dynamic analyzer **638** examines those packets which are initially allowed **632** by the ACL engine **621**. The dynamic analyzer **638** determines whether a given packet may require modification, due to embedded address or port number information. The dynamic analyzer **638** then separates packets which may require modification **640** from packets which do not require modification **639**. Packets which include IP address or port number information are identified by reading a protocol-specific field in the header. The dynamic analyzer **638** allows those initially allowed packets **632** and **635** which do not require modification **639** to pass through the firewall **642** into the private network.

The packets **640** which may require modification are then passed to an application-specific handler **641**. The application-specific handler **641**, as its name suggests, processes packets **640** for a particular application, such as FTP or NFS. The application-specific handler examines the protocol, session, port number and IP address, as well as the payload. In one embodiment, the application-specific handler may modify certain packets, which have been allowed **632** and **635**. If the IP address or port number in the packet header have been changed, for an incoming packet, or must be changed, for an outgoing packet, the application-specific handler translates the payload to reflect the change. In another embodiment, multiple application-specific handlers **641** may be provided, to process packets for different applications. For example, the firewall may include both an FTP-specific handler and an NFS-specific handler.

In another embodiment, the application-specific handler **641** may include the capability to send a "reset" packet to inform the TCP processor sending the denied packets that the connection has been denied. The connection is thereby rejected, rather than merely dropped. The rejection will prevent the TCP processor sending the denied packets **636** from continuing to try to connect with the network, thereby avoiding wasted bandwidth.

In conjunction with the software functionality description provided in the present disclosure, an apparatus in accordance with the preferred embodiments may be programmed using methods known in the art as described, for example, in Francise et. al., *Professional Active Server Pages* 2.0, Wrox Press (1998), and Zaration, *Microsoft C*++6.0 *Programmer's Guide*, Microsoft Press (1998), the contents of each of which is hereby incorporated by reference into the present application.

While preferred embodiments of the invention have been described, these descriptions are merely illustrative and are not intended to limit the present invention. For example, while the preferred embodiment discusses primarily a hardware implementation of a firewall, the scope of the preferred embodiments is not so limited. Those skilled in the art will recognize that the disclosed software and methods are readily adaptable for broader network analysis applications.

What is claimed is:

1. A method, comprising:

establishing a set of rules for controlling access to and from a network device for incoming and outgoing data units;

receiving, at the network device, a first sequence of data units; and

adding one or more first rules to the set of rules based on data extracted from the received first sequence of data units.

2. The method of claim **1**, further comprising:

filtering other data units received at the network device based on the set of rules.

3. The method of claim **2**, wherein filtering the other data units received at the network device comprises:

determining whether the other data units match one of the rules of the set of rules;

denying access to any of the other data units that do not match one of the rules of the set of rules; and

permitting access to any of the other data units that match one of the rules of the set of rules.

4. The method of claim **1**, wherein the network device is coupled to a private network and to a public network and the method further comprises:

receiving data units from the private network, wherein headers associated with the data units include network addresses and port numbers associated with source nodes in the private network; and

replacing, in the headers, the network addresses and port numbers associated with the source nodes with a network address and a port number associated with a firewall implemented at the network device.

5. The method of claim **4**, further comprising:

identifying data units of the received data units that have network addresses and port numbers associated with the source nodes embedded in payloads of the data units; and

replacing the network addresses and port numbers embedded in the payloads of the identified data units with the network address and a port number associated with the firewall.

6. The method of claim **1**, wherein the network device is coupled to a private network and to a public network and the method further comprises:

receiving data units from the public network, wherein headers associated with the data units include a network address and port number associated with a firewall implemented at the network device; and

replacing, in the headers for each of the data units, the network address and port number associated with the firewall with network addresses and port numbers associated with corresponding destination nodes in the private network.

7. The method of claim **6**, further comprising:

identifying data units of the received data units that have a network address and port number associated with the firewall embedded in payloads of the data units; and

replacing the network address and port number embedded in the payloads of the identified data units with the network addresses and port numbers associated with corresponding destination nodes for each of the data units in the private network.

8. The method of claim **1**, further comprising:

receiving a second sequence of data units; and

modifying the set of rules based on data extracted from the received second sequence of data units.

9. The method of claim **1**, further comprising:

receiving a second sequence of data units; and

adding one or more second rules to the set of rules based on data extracted from the received second sequence of data units.

10. The method of claim **1**, wherein the first sequence of data units comprises file transfer protocol (FTP) data units.

11. The method of claim **1**, wherein the data extracted from the received first sequence of data units comprises a port number and a network address associated with a destination of the data units.

12. The method of claim **1**, wherein the data extracted from the received first sequence of data units comprises a port number and a network address associated with a source of the data units.

13. A network device, comprising:

an access control engine configured to establish a set of rules for controlling access to and from the network device for incoming and outgoing data units; and

a dynamic filter configured to add one or more first rules to the set of rules based on data extracted from a first sequence of data units received at the network device.

14. A method for implementing a firewall at a network device, comprising:

receiving a first sequence of data units at the network device;

comparing the first sequence of data units to a first rule set to produce first comparison results;

comparing, based on the first comparison results, the received first sequence of data units with a second rule set to produce second comparison results;

filtering the first sequence of data units based on the second comparison results;

receiving a second sequence of data units;

extracting data from the received second sequence of data units; and

modifying the second rule set based on the extracted data.

15. The method of claim **14**, wherein filtering the first sequence of data units comprises:

denying access to any of the data units that do not match one of the rules of the second rule set; and

permitting access to any of the data units that match one of the rules of the second rule set.

16. The method of claim **14**, wherein the second sequence of data units comprises file transfer protocol (FTP) data units.

17. The method of claim **14**, wherein the data extracted from the received second sequence of data units comprises a port number and a network address associated with a destination of the data units.

18. The method of claim **14**, wherein the data extracted from the received second sequence of data units comprises a port number and a network address associated with a source of the data units.

19. The method of claim **14**, further comprising:

receiving a third sequence of data units at the network device;

comparing the third sequence of data units to the first rule set to produce third comparison results;

comparing, based on the third comparison results, the received third sequence of data units with the modified second rule set to produce fourth comparison results; and

filtering the third sequence of data units based on the fourth comparison results.

20. The method of claim **19**, wherein filtering the third sequence of data units comprises:

denying access to any of the third sequence of data units that do not match one of the rules of the modified second rule set; and

permitting access to any of the third sequence of data units that match one of the rules of the modified second rule set.

21. A network device, comprising:

an access control engine configured to compare a first sequence of data units received at the network device to a first rule set to produce first comparison results; and

a dynamic filter configured to:

compare, based on the first comparison results, the received data units with a second rule set to produce second comparison results,

filter the first sequence of data units based on the second comparison results, and

modify the second rule set based on data extracted from a second sequence of data units.

22. A method, comprising:

establishing a set of rules for controlling access to and from a network device for incoming and outgoing data units;

dynamically modifying the set of rules based on data extracted from first data units received at the network device; and

filtering second data units received at the network device based on the modified set of rules.

23. The method of claim **22**, wherein filtering the second data units received at the network device comprises:

determining whether the received second data units match one of the rules of the modified set of rules;

denying access to any of the received second data units that do not match one of the rules of the modified set of rules; and

permitting access to any of the received second data units that match one of the rules of the modified set of rules.

24. The method of claim **22**, wherein the second sequence of data units comprises file transfer protocol (FTP) data units.

25. The method of claim **22**, wherein the data extracted from the first data units comprises a port number and a network address associated with a destination of the first data units.

26. The method of claim **22**, wherein the data extracted from the first data units comprises a port number and a network address associated with a source of the first data units.

27. A network device, comprising:

an access control engine configured to establish a set of rules for controlling access to and from the network device for incoming and outgoing data units; and

a dynamic filter configured to:

modify the set of rules based on data extracted from first data units received at the network device to produce a first modified set of rules,

filter second data units received at the network device based on the first modified set of rules,

modify the first modified set of rules based on data extracted from second data units received at the network device to produce a second modified set of rules, and

filter third data units received at the network device based on the second modified set of rules.

28. A system, comprising:

means for establishing a set of rules for controlling access to and from a network device for incoming and outgoing data units;

means for dynamically modifying the set of rules based on data extracted from first data units received at the network device; and

means for filtering second data units received at the network device based on the dynamically modified set of rules.

* * * * *